# OPERATOR - SPLIT SOLUTION OF ADVECTION - DIFFUSION EQUATION IN COMPLEX REGIONS USING DOMAIN DECOMPOSITION

A Thesis Submitted
in Partial Fulfillment of the Requirements
for the Degree of

M A S T E R  O F  T E C H N O L O G Y

by

B.V.NAGABHUSHANA RAO

to the
Department of Mechanical Engineering
INDIAN INSTITUTE OF TECHNOLOGY, KANPUR
April, 1995

CERTIFICATE

It is certified that the work contained in the thesis entitled **OPERATOR - SPLIT SOLUTION OF ADVECTION - DIFFUSION EQUATION IN COMPLEX REGIONS USING DOMAIN DECOMPOSITION,** by B.V.NAGABHUSHANA RAO, has been carried out under my supervision and that this has not been submitted elsewhere for a degree.

K Muralidhar

K.MURALIDHAR
Dept. of Mechanical Engg
I.I.T Kanpur

April, 1995

ACKNOWLEDGEMENTS

I express my deep sense of gratitude to my guide Dr. K. Muralidhar for the untiring guidance and unceasing encouragement without which work wouldn't have been possible.

I would like to express my thanks to K.Mahesh, P.M.V. Subbarao, R.Venugopala Rao, TAN, Atul and Shejul for making my stay at I.I.T. Kanpur memorable.

I would like to thank my parents and siblings for their tremendous encouragement in all my endeavors.

*B.V.Nagabhushana Rao*

CONTENTS

LIST OF FIGURES

LIST OF TABLES

# SYNOPSIS

A numerical study of heat transfer from an array of cylinders placed in a flow field is presented. Novel techniques such as Operator - splitting and Domain Decomposition have been used for solving the governing differential equations. Galerkin Finite Element Method has been used for accommodating complex geometries.

The main source of error in the solution of heat transfer problems occurs in approximating the advective term (U∘VT) at high Peclet numbers. In the Operator Splitting algorithm the diffusion and advection terms are isolated from the complete equation and solved as accurately as possible. Operator Splitting is completely free of upwinding and hence free of false diffusion. This algorithm is valid for any Peclet number but at high Peclet numbers a high order interpolations will give oscillations in the temperature field. These have been removed in the present work by the Universal Limiters Method. Operator Splitting has a second advantage, i.e., it gives reasonable solutions even on coarse grid. This reduces the computational time.

Practical engineering problems inevitably involve intensive computations because of large size of physical domains and their complex geometries. In many cases computations cannot be implemented on existing computers due to limitations of memory. The current approach towards handling complex problems is to improve the numerical algorithms on one hand and computer performance on the other. With the advent of parallel computers there exists a great potential in increasing computer performance in terms of speed as well as memory. Domain Decomposition Techniques are the simplest of parallel algorithms that can

convert a sequential procedure to one suitable for parallel computing. Besides this Domain Decomposition has additional advantages which include generation of small matrices and marginal reduction in CPU time even on sequential machines. Implicit Domain Decomposition and Uzawa's algorithm has been used in the present work.

The model problem taken for analysis is the cooling of an array of heated cylinders buried in a porous medium, the earth for example. Transient and steady state results have been obtained for a Peclet number range of 0.1 to 100. They have been compared with analytical solutions whenever possible.

# CHAPTER 1

# INTRODUCTION

## 1.1 IMPORTANCE OF PRESENT WORK

Heat and mass transfer, fluid flow, chemical reactions and other related processes occur in engineering equipment, in the natural environment, and in living organisms. These processes play a vital role in a great variety of practical situations. Nearly all methods of power production involve fluid flow and heat transfer as essential processes. Major segments of chemical and metallurgical industries use components such as furnaces, heat exchangers, condensers and reactors, where thermofluid processes are at work. Forces that sustain motion of aircraft and rockets arise from fluid flow, heat transfer and chemical reaction. In the design of electrical machinery and electronic circuits, heat transfer is often a limiting factor. The spread of pollution in the natural environment is governed by principles of flow, heat and mass transfer.

The prediction of performance of a given physical system consists of finding the values of the relevant variables such as velocity, pressure, temperature and concentration of the chemical species in the process of interest at various instants of time. The predictions should also describe the effect of changes in geometry, flow rates, temperature and fluid properties on the performance of the system. Heat transfer and fluid flow processes can be studied either from laboratory experiments or by theoretical modelling. Experimental studies involve tests conducted on a full scale or scale-down model of the physical system. However, these tests are in most cases expensive and time consuming. Further the small scale models do not always simulate all the features of the full scale physical situation. For some physical systems, it is not possible to conduct model tests, for example, combustion in IC engines.

A theoretical calculation uses a mathematical model consisting of a set of differential equations that describe the physical system at every point inside the domain and at every point

in time. Very often, an analytical solution of these equations for a practical problem can not be found, thereby limiting its use for an engineer. Fortunately, the development of grid based numerical methods such as the finite element method and finite difference method and the spectacular progress in recent years in the speed of digital computers hold promise that the mathematical model of almost any practical problem can be worked out. This new methodology for attacking the complex problems in fluid mechanics and heat transfer has become known as Computational Fluid Dynamics and Heat Transfer. The main advantages of such a theoretical approach are low cost, remarkable speed with which the numerical investigation can be performed, complete information of all relevant variables and flexibility to simulate realistic conditions of the user's choice.

Convection-diffusion equations, arising in heat and mass transfer is an important class of problems that is being addressed in Computational Fluid Dynamics and Heat Transfer (CFDHT). They make appearance in many diverse forms such as Navier-Stokes equations with a specified pressure field, vorticity transport equation, energy balance equation and chemical transport equation. There are many numerical techniques such as finite difference method, Petrov-Galerkin finite element method and spectral method that are available for solving such equations.

For more than two-decades the numerical solution of convective diffusive transport phenomena has been a very active area research for finite difference and finite element modellers. The finite element method is very useful for handling complex geometries, especially in comparison to the finite difference method. The standard Bubnov-Galerkin finite element approach gives rise to spurious oscillations when transport is convection dominated. These oscillations result from difficulties associated with both the spatial and temporal discretization. The main difficulties arising in the numerical solution of the convection diffusion equation are due to the non-self adjoint character of the associated differential operator. The Galerkin method which is well suited to self adjoint problems leads to non-physical spatial oscillations when applied to highly convective flows. In such problems the convective contribution to the stiffness matrix makes

it non-symmetric. For getting relatively accurate answer this calls for fine meshes and better (but expensive) matrix inverters.

For complex geometries finite difference method is difficult to use, though the development in finite volume methods circumvents the problem to some extent. The finite element method is a better choice in this regard. In the finite element context most of the efforts to solve convection-diffusion problems have used upwind formulations introduced via Petro-Galerkin methods. It has now been realized that like their finite difference counterparts finite element upwind procedures tend to produce over diffusive solutions in multidimensional and transient situations. One more disadvantage of upwinding in finite element method (FEM) is the problem of selecting proper trial and test functions.

The above difficulties can be overcome at least partially by the operator splitting (OS) method. In the OS formulation, we eliminate the convective contribution in Bubnov-Galerkin finite element formulation. This ensures that inversion of highly unsymmetric matrix is avoided. The convective term, source and decay terms if they are present are solved separately. The method of characteristics is used to handle the effect of convection alone. Even in the OS method, one gets some undershoots or overshoots in the transported variable in highly convective flows due to interpolation errors. In the present work the method of universal limiters (B.P. Leonard et al, 1990) is used for removing interpolation related oscillations.

The present work is concerned with two problems, namely, one dimensional unsteady transport and heat transfer from an array of heated cylinders placed in a flow field. At high Peclet numbers a refined mesh is required over large domains. This necessarily leads to the requirement of a large computer memory. The memory problem has been resolved in the present study using a domain decomposition technique.

## 1.2 INTRODUCTION TO OPERATOR SPLITTING

The operator splitting algorithm (to be called OS) is a special case of splitting methods or fractional step methods which reduces the solution of a complicated problem to a successive

solution of simpler problems. Various kinds of splittings are possible. These are given below :

(a) Geometrical Splitting: This reduces a multidimensional problem to a temporal sequence of problems of smaller dimensions and in particular to transient one dimensional problems. In the latter case we shall speak of splitting in terms of spatial direction. A.D.I. (alternating direction implicit) scheme for the solution of the transient two dimensional heat equation is an example of this kind.

(b) Physical Splitting: Here the original physical process is represented as a temporal sequence of processes possessing simpler physical structures. This kind of splitting can be treated as splitting in terms of the physical processes.

(c) Analytical Splitting: This allows various analytical problems to be solved in fractional steps. For example, the restoration of divergence in predictor corrector scheme or treating the algebraic terms of the original equations as a separate operator in curvilinear coordinates.

The OS algorithm belongs to the splitting of type (c). This algorithm properly identifies the mixed mathematical character of the governing differential equation and solves each homogeneous component of the equation as accurately as possible. Hence the OS algorithm solves the governing equation true to its character.

A typical convection-diffusion equation arising in heat transfer problems is of the form.

$$\frac{\partial T}{\partial t} + ( u . \nabla T ) = \alpha \nabla^2 T \qquad (1.1)$$

In cartesian coordinates $\nabla = ( \frac{\partial}{\partial x} , \frac{\partial}{\partial y} )$, u is velocity vector, T is temperature and $\alpha$ is thermal diffusivity. The mathematical character of the equation is elliptic if $t \longrightarrow \infty$ parabolic if u is small and hyperbolic if u is large in magnitude. In all other cases, the equation is said to be mixed in character. These special cases are summarized below:

Steady state $(t \longrightarrow \infty)$ $(\tilde{u}.\tilde{\nabla}T) = \alpha \nabla^2 T$ (elliptic) (1.2)

Conduction limit $(|\tilde{u}| \longrightarrow 0)$ $T_t = \alpha \nabla^2 T$ (parabolic) (1.3)

Convection limit $(|\tilde{u}| \longrightarrow \infty)$ $T_t + \tilde{u}.\tilde{\nabla}T = 0$ (hyperbolic) (1.4)

The OS algorithm treats this problem as follows. At any time

level t, one solves Equation (1.4) over a time step $\Delta t$. This is considered as a predictor step. The corrector step (Equation 1.3) is solved over the same time interval $\Delta t$ with the solution obtained in the first step as the initial condition. At high values of velocity, the correction required is quite small and at low velocities, the predictor step acting alone is inadequate. However, the two steps confined will furnish the complete solution of Equation (1.1) irrespective of the magnitude of u.

The above approach is completely free of upwinding and hence avoids false diffusion errors that usually occur at high Peclet numbers. The only error arising in the O.S. algorithm is due to time discretization and it is of order $\Delta t$. If either the predictor step or the corrector step is solved numerically on a grid size of $\Delta t$, then additional truncation errors ($\sim \Delta x^n$, $n \geq 1$) are possible. Hence the overall error is

$$e \sim O\ (\Delta x^n,\ \Delta t),\ n \geq 1$$

where n depends on the scheme used to solve the diffusive corrector step. Numerical techniques are well established for solving the diffusion equation. This equation does not have an analytical solution when solved in a complex geometry. the finite element method is most appropriate for this purpose. In finite element method the discretization error depends on interpolation function used in the problem.

The main source of error in solving heat transfer problems occurs while approximating the advection terms $(u.\nabla T)$. In the OS algorithm, these terms are isolated from the complete equation and solved as accurately as possible.

On non-dimensionalization, the heat transfer equation becomes

$$T_t + (u.\nabla T) = \frac{1}{Pe} \nabla^2 T \qquad (1.5)$$

where $Pe = UL/\alpha$, is called as Peclet number. It is a measure of the relative importance of convective transport with respect to diffusive transport. Most of the existing algorithms fail or become inaccurate when Pe is raised. The OS algorithm proposed here is uniformly valid over any range of Pe. As the Peclet Number is raised, a discontinuity in the initial conditions will propagate through the flow domain unchanged. This is called a front. Predicting the occurrence of a front and its location is a

challenging numerical problem. Fronts can also form in non-linear problems at low or moderate Peclet numbers.

The OS algorithm is only conditionally stable since it involves explicit time marching and is limited by the Courant stability principle. The stability criterion is stated in words as, "the numerical movement of the front should not exceed one grid spacing in one time increment since the maximum front speed is the fluid velocity. Hence, we require $u \frac{\Delta t}{\Delta x} \leq 1$, for stability.

## 1.3 INTRODUCTION TO DOMAIN DECOMPOSITION

Practical engineering problems inevitably involve intensive computations because of complex geometries and large domains. The numerical solution using finite element approximations of complicated two and three dimensional nonlinear problems can be a formidable task. The difficulties encountered here are computing time and memory space. In many cases the computations cannot be implemented on existing computers at all. The current approach towards handling complex problems is to improve the numerical algorithm on one hand and computer performance on the other hand. With the advent of parallel computers there exists a great potential in increasing computer speeds as well as memory. Parallel computers require parallel algorithm to exploit the distributed nature of the processor architecture. Domain decomposition techniques are attractive because they can mix direct solvers used at the subdomain level and iterative solvers used at interface level. Moreover these techniques are ideally suited to modern parallel computers, because of the built-in parallelism of the algorithm and good localization of the associated data. They are fully equivalent to the matrix partitioning approach, but have the advantage that subdomains can be conveniently chosen based on the problem geometry. Besides the ability to parallelize, other specific advantages of domain decomposition are given below:

(1) It can generate smaller matrices that require less memory and are rapidly invertible when iterative methods are used.

(2) A complex assemblage of components of system can be systematically analyzed.

(3) Complex phenomena can be localized by assigning separate subdomains to them.

The basic theme of domain decomposition (DD) is that a large domain is divided into many subdomains each of which is easily described. The subdomains are linked at the interfaces. The governing equations are then solved independently in each subdomain. The individual solutions are assembled carefully to get a converged solution of the original problem on the complex physical domain. The efficiency and accuracy of domain decomposition lies in its interface treatment. The subdomains in practice may be overlapping or non-overlapping.

The domain splitting methods originate from well known Schwarz method (Le Tallec,1994) for solving elliptic problems. First we consider non-overlapping technique with model problem.

Consider a linear governing differential equation of the type

$$\Delta u = f \quad \text{in } \Omega \tag{1.6}$$

and $u = g$ on $\partial\Omega$

where f and g are known functions. Here $\Omega$ is the physical region and $\partial\Omega$ is its boundary as shown in Figure 1.1. $\Delta$ is the Laplacian operator. Figure 1.2 shows division of the full domain into two subdomains 1 and 2. The interface between the two subdomains is denoted as 'C'. We seek a solution of Equation (1.6) in the original large domain. The problem in region 1 is solved numerically with an arbitrary Dirichlet boundary condition on C. Let this solution be $u_1^1$, u being the original dependent variable. This is followed by numerical solution of u in region 2 with a boundary condition on C which is derived from $u_1^1$. The global solution u and its derivative should be continuous at the interface. Hence the most obvious choice of boundary condition on C for region 2 would be normal derivative $\frac{\partial u_1^1}{\partial n}$. Here n denotes the normal at C exterior to region 1. With this boundary condition u can be solved in region 2. Let this solution be $u_2^1$. In mathematical form, we have in region 1

$$\Delta u_1 = f \text{ in } \Omega_1 \tag{1.7}$$

$$u_1 = g \text{ on } \partial\Omega_1 \text{ and } u_1 = \lambda \text{ on C}$$

In region 2

$$\Delta u_2 = f \text{ in } \Omega_2$$

$$u_2 = g \text{ on } \partial\Omega_2 \text{ and } \frac{\partial u_2}{\partial n_2} = - \left. \frac{\partial u_1}{\partial n_1} \right|_{\text{at C}}$$

Here $\partial\Omega = \partial\Omega_1 \cup \partial\Omega_2$ and $\Omega \sim \Omega_1 \cup \Omega_2$

Parameter $\lambda$ is the Dirichlet boundary condition for $u_1$ described earlier and $n_1$ and $n_2$ are the normals at C exterior to regions 1 and 2 respectively.

The problem in region 1 is solved again with a new boundary condition on C. This may be the value of $u_2^1$ on C or some combination of $u_2^1$ and $u_1^1$ on C. Let the solution be denoted $u_1^2$. The procedure continues in this fashion where the problem in regions 1 and 2 are solved alternately until convergence in the global solution is achieved. This convergence criterion will satisfy the conditions of continuity of u and its derivative at the interface. The method described above is known as the alternating Dirichlet-Neumann (D-N) Method because Dirichlet and Neumann boundary conditions are applied alternatively on the interface.

The domain decomposition method described above is sequential in nature. Its efficiency depends on how the iterations are handled at the subdomain interfaces. Consider the alternating D-N method as an example. After one complete sweep of the domain, the use of arithmetic means of $u_1^I$ and $u_2^I$ as the boundary condition on C for $u_1^{I+1}$ can lead to a considerable improvement in overall convergence rate. Here notation I represents iteration number. Thus the interfaces are the most crucial zones and they should be handled with care and intelligence. In the present work, Uzawa's algorithm has been used for interface iterations.

## 1.4 LITERATURE SURVEY

The operator splitting technique, which is being presented here as an alternative to the upwind method was first proposed by Yanenko et al. (1984). Various possible types of operator splittings feasible in gas dynamics equations have been described in their work. Issa (1985) describes a non iterative method for

handling the coupling of the implicitly discretized time dependent fluid flow equations. The method is based on the use of pressure and velocity as dependent variables and hence applicable to both compressible and incompressible versions of the momentum equations. The main feature of the technique is the splitting of the solution process into a series of steps whereby operations on pressures are decoupled from those on velocity at each step, with split sets of equations being amenable to solution by standard techniques. Cooke (1986) has mathematically established that a certain operator splitting of the two dimensional Navier-Stokes equations in conservative form is second order accurate not only for linear systems, but also is true in the presence of non-linearity. Glowinksi et al. (1992) describe various operator splitting methods for advection - diffusion equations and their drawbacks.

Ding and Liu (1989) present a new operator splitting algorithm for the two dimensional convection-dispersion-reaction equation. The governing equation is split into three successive initial value problems: a pure convection problem, a pure dispersion problem and a pure reaction problem. For the pure convection problem solutions are found by a high order upwind procedure. For the pure dispersion problem, time-explicit finite element algorithm is employed. Analytical solutions are obtained for the pure reaction problem. Results presented are mainly for one dimensional problems.

Muralidhar et al. (1993) have applied the operator splitting algorithm to two dimensional problems in simple geometries and describe the advantages of operator splitting over upwinding. They have subsequently applied it to enhanced oil recovery using hot water injection (Muralidhar and Pillai, 1993).

Leonard and Mokhtari (1990) present a method for removing oscillations encountered in high order upwinding schemes. Results presented are mainly for one-dimensional problems. Leonard and Nikhafs (1991) discuss a strategy for accurately simulating highly convective flows containing discontinuities such as density fronts or shock waves, without distorting the smooth profiles or clipping narrows local extreme and disadvantages of other shock capturing methods. This method has also been applied only to

one-dimensional problems. Westerink and Shea (1989) explain how to handle oscillations arising due to strong convection in finite element method through the use of a Petrov-Galerkin method.

Glowinksi et al (1983) discuss domain decomposition methods for nonlinear problems in fluid dynamics in complex geometries using finite element method. They have given a summary of domain decomposition methods suitable for parallelization using Schwarz's algorithm.

Yagawa et al (1991) have developed a parallel finite element method based on domain decomposition for stress analysis. They use Uzawa's algorithm and the conjugate gradient method for interface treatment. Chatterjee (1993) has applied domain decomposition technique to oil recovery and transient non-linear heat conduction. He uses Uzawa's algorithm for interface treatment.

Laura and Habashi (1994) present the parallelization aspects of a solver for fully coupled 3-D compressible Navier—Stokes equations. Finite element method is used for solving the differential equations and conjugate gradient method is used for interface treatment. They explain the difficulties encountered in domain decomposition methods owing to book keeping of geometry and convergence criterion at interfaces for non-linear problems.

## 1.5 SCOPE OF THE PRESENT WORK

The present work is in two parts. In the first part the performance of operator splitting for advection-diffusion problem is tested for simple as well as complex geometries. In the present work, the velocity field is calculated using Darcy's law applied to flow in a porous medium. With operator splitting, the diffusion equation is solved using finite element method and the advection step is solved using method of characteristics. The results obtained with operator splitting are compared with analytical results whereever possible. To remove errors due to higher order interpolation, universal limiters have been used.

In the second part of this study, the performance of domain decomposition is tested for simple as well as complex geometries. Only diffusion step is solved locally in each subdomain, and

advection step is solved globally over the whole domain. Uzawa's algorithm is used for interface treatment. The effect of location of outflow plane has been studied using two subdomains. In the present work, the original domain is divided into 2, 4 and 9 subdomains and their performance has been studied.

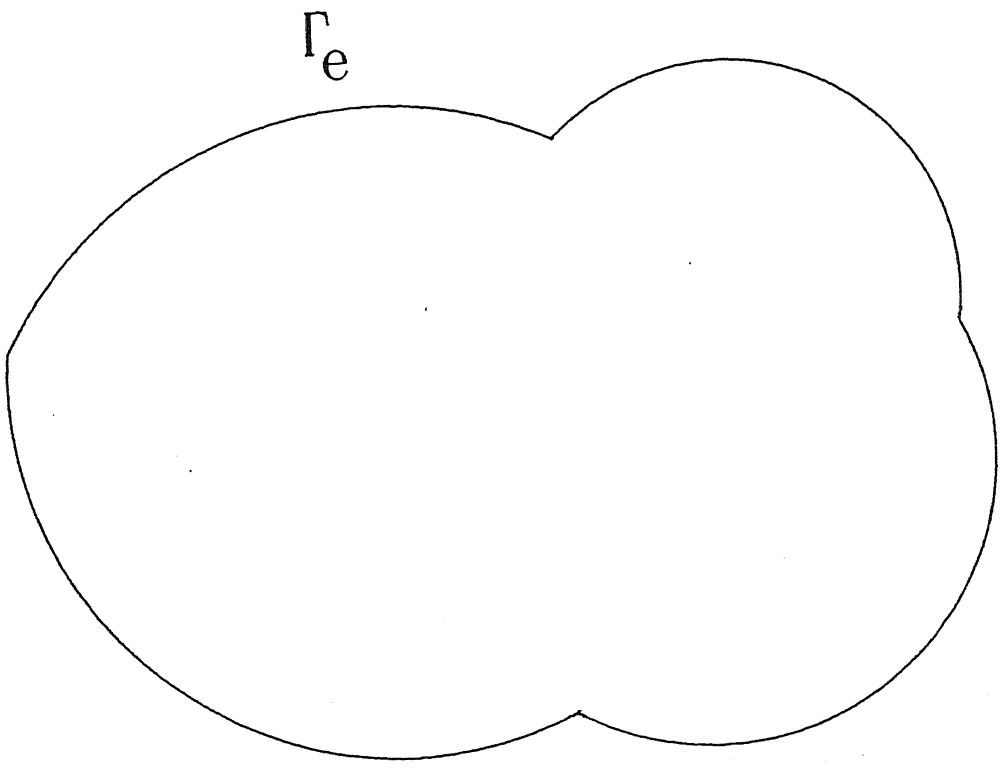



Figure 1.1 Schematic Diagram of Computational Domain

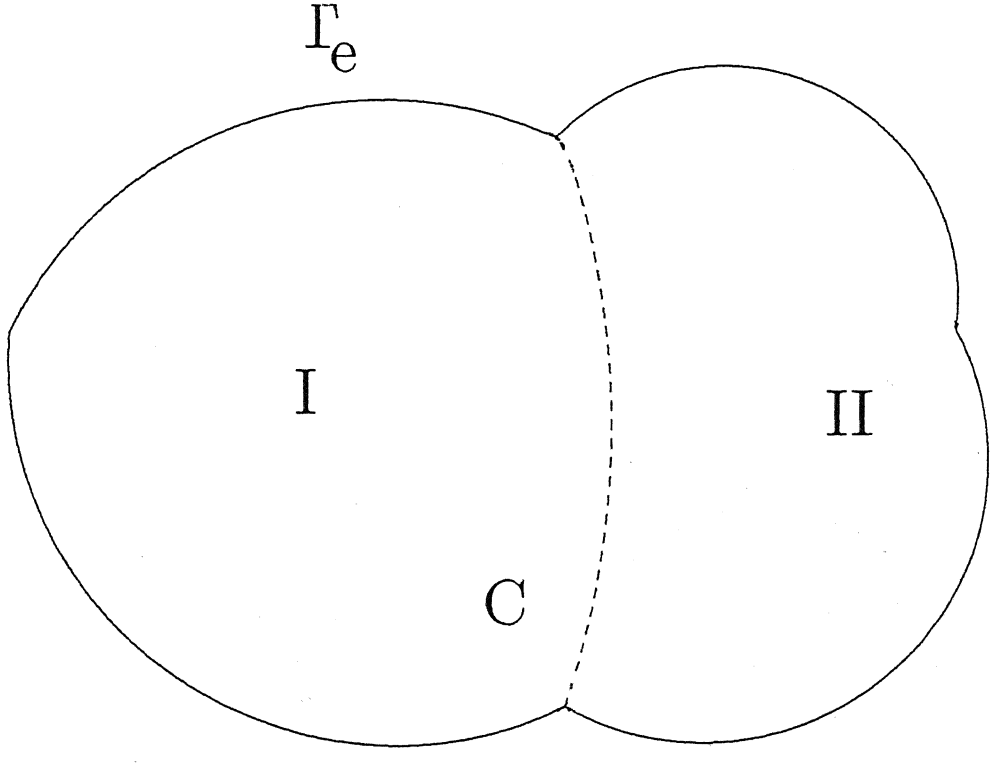



Figure 1.2 Computational Domain divided into Two Subdomains

# CHAPTER 2

# MATHEMATICAL FORMULATION

## 2.1 GOVERNING DIFFERENTIAL EQUATIONS, BOUNDARY CONDITIONS AND INITIAL CONDITIONS

Consider the differential equation governing energy transport,

$$T_t + u.\nabla T = \alpha \nabla^2 T \qquad (2.1)$$

where T is temperature, u is velocity vector and $\alpha$ is thermal diffusivity. Non-dimensionalization of Equation (2.1) is carried out by using R as the length scale, $R^2/\alpha$ as time scale, approach velocity U as the velocity scale and qR as the temperature scale. For flow past a cylinder or an array of cylinders, R is the cylinder radius and q is the specified heat flux on its surface. In a problem without cylinders, R is a longitudinal distance and the temperature scale is taken as $T_{max} - T_{min}$.

Equation (2.1) in dimensionless form is,

$$T_t + u.\nabla T = \frac{1}{Pe} \nabla^2 T \qquad (2.2)$$

where $Pe = UR/\alpha$ is Peclet number.

The boundary conditions chosen for the present study are given as follows.

Case I: Movement of a temperature front in a semi infinite medium

$$x = 0,\ 0 \le y \le L;\ T = 1$$

$$x = L,\ 0 \le y \le L;\ T = 0$$

$$y = 0 \text{ and } L,\ 0 \le x \le L,\ \frac{\partial T}{\partial n} = 0$$

where n is a unit outward normal and L is length scale.

The initial condition is taken as T = 0 at t = 0, i.e., the medium is initially cold. We study the response of the medium when it is subject to hot fluid suddenly at its inflow plane.

Case II: Flow Past an Array of Heated Cylinders

$$x = 0,\ 0 \le y \le L;\ \ T = 0$$

$$x = L,\ 0 \le y \le L;\ \frac{\partial T}{\partial n} = 0$$

$y = 0$ and $L$, $0 \leq x \leq L$; $\frac{\partial T}{\partial n} = 0$

On the surface of the cylinders, $\frac{\partial T}{\partial n} = -1$

where n is a unit outward drawn normal.

The initial condition is $T = 0$, at $t = 0$, as before. The velocity field in both problems is steady and prescribed.

## 2.2 OPERATOR SPLITTING FORMULATION

The OS algorithm properly identifies the mixed mathematical characters of the governing differential equation and solves each equation as accurately as possible.

Consider the governing differential equation for energy transport in 2-dimensional

$$T_t + u.\nabla T = \frac{1}{Pe} \nabla^2 T \tag{2.3}$$

Application of the OS algorithm to equation (2.3) generates the following steps.

$$T_t = \frac{1}{Pe} \nabla^2 T \tag{2.4}$$

$$T_t + u.\nabla T = 0 \tag{2.5}$$

At any time interval, we solve equation (2.4) over a time step $\Delta t$. This is the predictor step. The corrector step is subsequently solved over the same time interval $\Delta t$ with the solution of the predictor step as the initial condition. The two steps combined will completely solve the full equation irrespective of the magnitude of Peclet number. In the present study, the predictor step is solved using a Bubnov-Galerkin finite element method. The finite element method formulation is described in Appendix A. The corrector step is solved using the method of characteristics and is described in Appendix B.

## 2.3 UNIVERSAL LIMITERS

Accurate simulation of highly convective flows continues to be one of the most challenging problems in computational mechanics. The inability to adequately simulate simple scalar profiles even in straight forward case of 1-dimensional pure convection at constant velocity has been referred to as the ultimate embarrassment for computational mechanics. All existing

methods are well suited for handling smooth solutions. A sudden change in the value of dependent variable gives spurious unphysical oscillations.

In the present work no upwinding has been used. However, in the operator splitting algorithm a high order interpolation procedure is used for calculating the temperature. Interpolation methods of order two or higher are themselves a source of oscillations. To limit the oscillations, the method of universal limiters (Leonard & Niknafs, 1991) has been adopted in the present work and is described below.

Figure (2.1) shows the local behaviour of any convected variable $\phi$. Depending on the direction of the local velocity we label the three indicated node values $\phi_u$ (upwind), $\phi_D$(downwind) and $\phi_C$ (centrally located between $\phi_C$ and $\phi_D$). In Figure (2.1(a)) the local variation of $\phi$ is monotonic. One necessary condition of the universal limiter is sketched in this figure and is shown by the cross-hatched limits. The convected face value $\phi_f$, should lie between adjacent node values in locally monotonic regions.

The treatment of locally non-monotonic behaviour is shown in Figure (2.1) (c) and (d). Handling this case is more difficult compared to monotonic behaviour.

It is convenient to summarize the limiter constraints in terms of normalized variables as follows. Let

$$\tilde{\phi}(x,y,z) = \frac{\phi(x,y,z) - \phi_u}{\phi_D - \phi_u}$$

Hence $\tilde{\phi}_u = 0$, $\tilde{\phi}_D = 1$

Symbolically, the universal limiter constraints on $\tilde{\phi}_f$ can be written (in monotonic case) as,

$$\tilde{\phi}_c \le \tilde{\phi}_f \le 1, \quad \text{for } 0 \le \tilde{\phi}_c \le 1 \qquad (2.5)$$

and $0 \le \tilde{\phi}_u \le \tilde{\phi}_c^{\,n}$

This above condition is not enough to guarantee computational monotonicity. For this consider the explicit update of $\tilde{\phi}_c$ under conditions of constant velocity. In terms of normalized variables,

$$\tilde{\phi}_c^{\,n+1} = \tilde{\phi}_c^{\,n} - C\,(\tilde{\phi}_f - \tilde{\phi}_u)$$

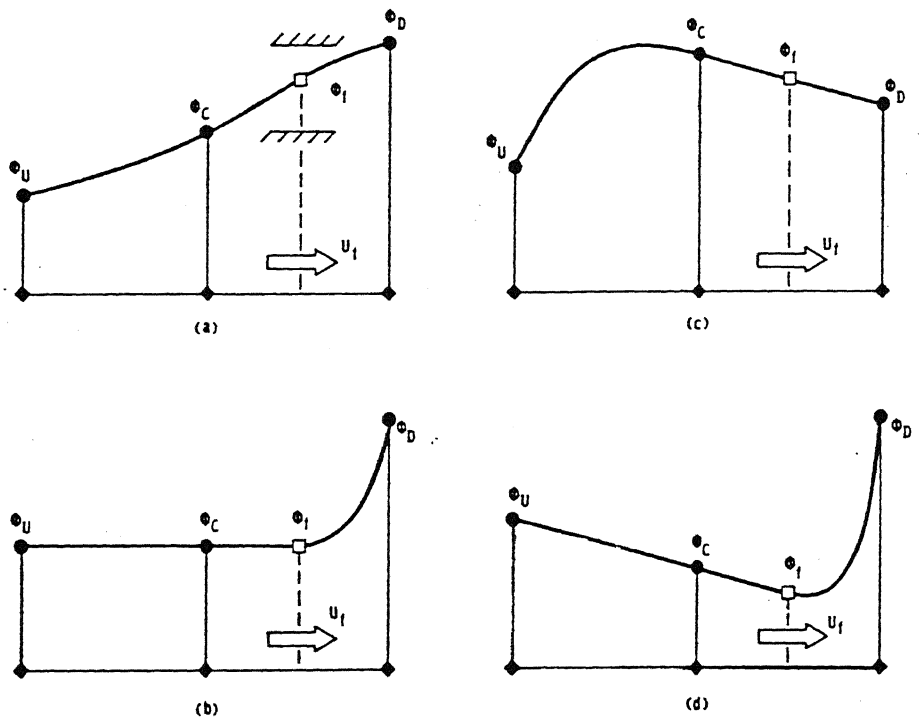



Figure 2.1 Selection of Universal Limiters : (a) and (b) Monotonic Variation in Variable $\phi$, (c) and (d) Non-monotonic Variation in $\phi$

where $C = \frac{U\Delta t}{\Delta x}$ is the Courant number.

In order to maintain monotonicity (locally), $\tilde{\phi}_c$ must satisfy

$$\tilde{\phi}_u^{n+1} \le \tilde{\phi}_c^{n+1} \le \tilde{\phi}_D^{n+1}$$

Taking the conservative worst case estimate implies,

$$0 \le \tilde{\phi}_c^{n+1} \le 1$$

From Equation (2.5), the right hand side of the inequality is assured. The left hand inequality leads to,

$$\tilde{\phi}_f \le \tilde{\phi}_u + \tilde{\phi}_c/C$$

Again the worst case estimate gives $\tilde{\phi}_u = 0$ and so,

$$\tilde{\phi}_f \le \tilde{\phi}_c/C \qquad (2.6)$$

The inequalities Equation (2.5) and Equation (2.6) constitute the universal limiter constraints on $\tilde{\phi}_f$ with respect to $\tilde{\phi}_c$ when $\tilde{\phi}_c$ is within the monotonic range ($0 \le \tilde{\phi}_c^{n} \le 1$). Figure (2.2) shows a clear picture of these constraints. The algorithm for applying the method of universal limiters is as given below.

1. According to the direction of velocity identify $\phi_u$, $\phi_c$ and $\phi_D$.
2. Compute $\phi_f$ using some interpolation scheme, say cubic or quadratic.
3. Compute the corresponding normalized variables.
4. Check the limits on $\tilde{\phi}_f$ (Equation (2.5), and Equation (2.6)).
5. If $\tilde{\phi}_f$ falls within the region (Figure 2.2) skip the limiter's algorithm and go to 8, if not go to 6.
6. Choose the nearest possible value of $\tilde{\phi}_f$.
7. Denormalize $\tilde{\phi}_f$ with following expression

$$\phi_f = \tilde{\phi}_f\,(\phi_D - \phi_u) + \phi_u$$

8. Repeat this procedure for each node.

The above approach is given for a one-dimensional problem. In the two-dimensional case, the algorithm is implemented along a streamline assuming it to be locally straight.

## 2.4 FORMULATION FOR DOMAIN DECOMPOSITION

### 2.4.1 Uzawa's Algorithm with an Example

We first outline Uzawa's algorithm for an elliptic problem to

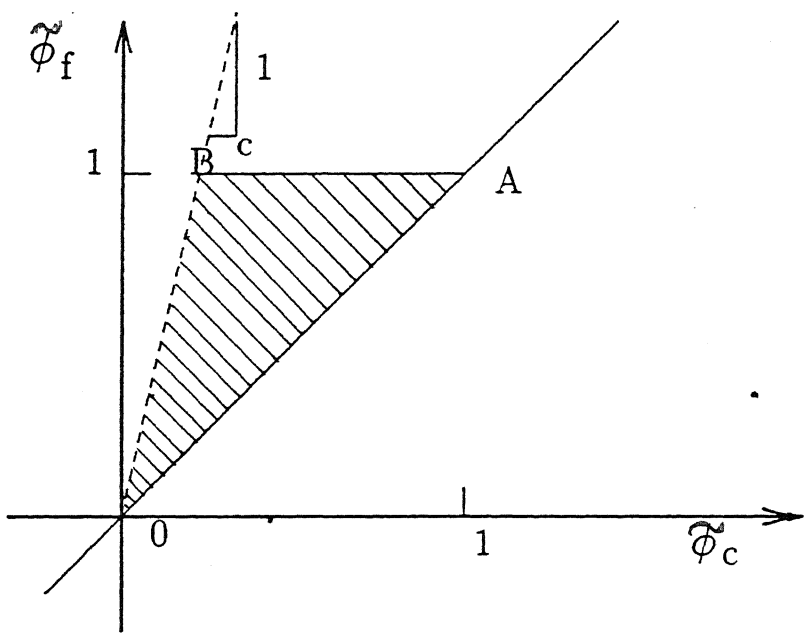



Figure 2.2 Universal Limiter Constraints in the Normalized Variable Diagram

show its parallel nature and hence provide a clear motivation for its use in the present study. Consider a linear second order partial differential equation of the type,

$$\Delta u = f \text{ in } \Omega \quad \text{and} \quad u = g \text{ on } \partial\Omega \tag{2.7}$$

Here f and g are known functions, $\Omega$ is the physical region and $\partial\Omega$ is its boundary (Figure 1.1). Figure 1.2 also shows a decomposition of $\Omega$ into two subdomains 1 and 2 with C as their interface. For a globally converged solution one requires both u and its normal derivatives to be continuous on C. We start with a guess value of the derivatives, $\frac{\partial u}{\partial n_1}\Big|_C$ and $\frac{\partial u}{\partial n_2}\Big|_C$ ( $= \lambda^1$) where $n_1$ and $n_2$ are the normals at C exterior to regions 1 and 2 respectively. Let this guess be $\lambda^1$. Using $\lambda^1$ as the boundary condition on C and solves the governing differential equation in regions 1 and 2. Let the solutions respectively be $u_1^1$ and $u_2^1$. Let $y_1^1$ and $y_2^2$ be the values of $u_1^1$ and $u_2^1$, respectively on C. Then $\lambda^1$ is updated as,

$$\lambda^2 = \lambda^1 + \rho\ (y_2^1 - y_1^1)/L$$

Here $\rho$ is an adjustable constant that lies between 0 and 1 and L is a suitable length scale. Using $\lambda^2$ as the next guess, the entire process is repeated. After I iterations, Uzawa's algorithm for updating the interface flux is,

$$\lambda^{I+1} = \lambda^I + \rho(y_2^I - y_1^I)/L \tag{2.8}$$

Interface convergence is judged by the criterion

$$|\lambda^{I+1} - \lambda^I| \leq \delta$$

where $\delta$ is the specified convergence criterion.

The iterations of the algorithm given above converge when the computed function $u^I(x)$ matches the steady state solution of Equation (2.7). In a transient problem, the same algorithm as applied between successive time steps and each subproblem in the subdomains are solved by an implicit finite element method. The choice of the implicit scheme guarantees the stability of the time-marching procedure and convergence is dictated by the magnitude of interface parameter, $\rho$ alone. When the gradient of the dependent variable converges, the variable at the interface becomes continuous and this is implied by Equation (2.8). Hence

the algorithm ensures continuity of u as well as its normal derivative at the interface. It is worth noting that iterations are completely avoided if the transient problem is solved by an explicit time-marching procedure.

It can be seen that Uzawa's algorithm given above is completely parallel in nature. Calculations in regions 1 and 2 can be carried out independently on two different processors of a parallel computer. The processors return the value of $y_1^I$ and $y_2^I$ after each iteration to the front end processor (FEP). The FEP manipulates the incoming data and performs convergence tests before sending the data back to the individual processors for subsequent calculations. In general, one can handle as many subdomains as the number of processors available.

Equation (2.8) shows that domain decomposition introduces interface iterations even in linear problems. Hence the possibility of an increase in CPU time must be considered, especially under transient conditions. However results presented by Chatterjee (1993) show that CPU time marginally decreases with increase in the number of subdomains even on sequential machines. The reason for this result is the following. When matrix inversion is accomplished using iterative methods or direct methods e.g. Gauss-Seidel or Gauss elimination; the convergence rate varies as $N^{+\alpha}$, where N is the matrix size and $\alpha$ is some positive constant. Hence convergence is significantly delayed for large values of N. Application of domain decomposition cuts the matrix size by a factor greater than or equal to 2, depending on the number of subdomains used. Hence each subdomain calculation is accelerated, providing an overall computing time advantage.

### 2.4.2 Movement of a Front: 2-subdomains

Governing differential equation: $T_t = \frac{1}{Pe} \nabla^2 T$

$0 < x < 18,\ 0 < y < 6,\quad t > 0$ (2.9)

Initial conditions: $T = 0$ at $t = 0$

Boundary conditions : $x = 0,\ 0 \le y \le 6;\ T = 1$

$x = 18,\ 0 \le y \le 6;\ T = 0$

$y = 0$ and $6,\ 0 \le x \le 18;\ \frac{\partial T}{\partial n} = 0$

As before, n is outward drawn normal on a surface. The region size 18 x 6 has been arbitrarily chosen in this work.

Figure (2.3) shows the decomposition of the original problem into two equal subdomains. While the interface can be chosen to be an arbitrary surface, it is most convenient to have it overlap with one of the coordinate lines. In the present example, it is the coordinate line x = 9. The mathematical problem solved in each subdomain are given below.

Subdomain 1

The governing differential equation: $T_t = \frac{1}{Pe} \nabla^2 T$

$0 < x < 9 , \quad 0 < y < 6, \quad t > 0$ (2.10)

Initial condition: $T = 0$ at $t = 0$,

Boundary conditions: $x = 0, \ 0 \le y \le 6; \ T = 1.0$

$x = 9, \ 0 \le y \le 6; \ \frac{\partial T}{\partial n} = - \lambda^I$

$y = 0$ and $6, \ 0 \le x \le 9; \ \frac{\partial T}{\partial n} = 0$

Subdomain 2

Governing differential equation: $T_t = \frac{1}{Pe} \nabla^2 T,$

$6 < x < 12, \quad 0 < y < 6, \quad t > 0$ (2.11)

Initial condition: $T = 0$ at $t = 0$,

Boundary conditions: $x = 6, \ 0 \le y \le 6; \ \frac{\partial T}{\partial n} = \lambda^I$

$x = 18, \ 0 \le y \le 6; \ T = 0$

$y = 0$ and $6, \ 9 \le x \le 18; \ \frac{\partial T}{\partial n} = 0$ .

Equations (2.10) and (2.11) are solved by an implicit Bubnov Galerkin finite element method using six-noded isoparametric elements. The stiffness matrix is inverted using a sparse matrix solver (MA28). The matrix solver uses Gaussian elimination technique with full pivoting. At the I-th iteration between successive time steps, let $T_1^I$ and $T_2^I$ be the temperature values computed at the interface from subdomain 1 and 2 respectively. Then, the guessed gradient boundary condition $\lambda$ is updated using Uzawa's algorithm as follows.

$$\lambda^{I+1} = \lambda^I + \rho \ (T_2{}^I - T_1{}^I)/L \qquad (2.12)$$

For a characteristic dimension of L = 1, $\rho$ = 0.2 has been found to yield optimum results in the earlier work of Chatterjee (1993). This value has been used again in this study. Figure (2.4) shows a flow chart for implementing Uzawa's algorithm. The convergence criteria used is $|\lambda^{I+1} - \lambda^{I}| \leq \delta$, where $\delta$ is a small value, typically $10^{-4}$ in dimensionless calculations.

### 2.4.3 Movement of a Front: 4-Subdomains

Governing differential equation: $T_t = \frac{1}{Pe} \nabla^2 T$

$0 < x < 18, \quad 0 < y < 6, \quad t > 0$ (2.13)

Initial condition: $T = 0$ at $t = 0$

Boundary conditions: $x = 0, \ 0 \leq y \leq 6; \ T = 1$

$x = 18, \ 0 \leq y \leq 6; \ T = 0$

$y = 0$ and $6, \ 0 \leq x \leq 18; \ \frac{\partial T}{\partial n} = 0$

Figure (2.5) shows the decomposition of the original problem into four equal subdomains. The mathematical problems solved in each subdomain are given below.

Subdomain 1

Governing differential equation: $T_t = \frac{1}{Pe} \nabla^2 T,$

$0 < x < 9, \quad 0 < y < 3, \quad t > 0$ (2.14)

Initial condition: $T = 0$ at $t = 0$

Boundary conditions: $x = 0, \ 0 \leq y \leq 3; \ T = 1$

$x = 9, \ 0 \leq y \leq 3; \ \frac{\partial T}{\partial n} = - \lambda_1^I$

$y = 0, \ 0 \leq x \leq 9; \ \frac{\partial T}{\partial n} = 0$

$y = 3, \ 0 \leq x \leq 9; \ \frac{\partial T}{\partial n} = -\lambda_4^I$

Subdomain 2

Governing differential equation: $T_t = \frac{1}{Pe} \nabla^2 T$

$9 < x < 18, \quad 0 < y < 3, \quad t > 0$ (2.15)

Initial condition: $T = 0$ at $t = 0$

Boundary conditions: $x = 9, \ 0 \leq y \leq 3; \ \frac{\partial T}{\partial n} = \lambda_1^I$

$x = 18,\ 0 \le y \le 3;\ T = 0$

$y = 0,\ \ 9 \le x \le 18;\ \frac{\partial T}{\partial n} = 0$

$y = 3,\ 9 \le x \le 18;\ \frac{\partial T}{\partial n} = -\lambda_2^I$

Subdomain 3

Governing differential equation: $T_t = \frac{1}{Pe} \nabla^2 T$

$$0 < x < 9,\ \ 3 < y < 6,\ \ t > 0 \qquad (2.16)$$

Initial condition: $T = 0$ at $t = 0$

Boundary conditions: $x = 0,\ 3 \le y \le 6;\ T = 1$

$x = 9,\ 3 \le y \le 6;\ \frac{\partial T}{\partial n} = -\lambda_3^I$

$y = 3,\ 0 \le x \le 9;\ \frac{\partial T}{\partial n} = \lambda_4^I$

$y = 6,\ 0 \le x \le 9;\ \frac{\partial T}{\partial n} = 0$

Subdomain 4

Governing differential equation: $T_t = \frac{1}{Pe} \nabla^2 T$

$$9 < x < 18,\ \ 3 < y < 6,\ \ t > 0 \qquad (2.17)$$

Initial condition: $T = 0$ at $t = 0$

Boundary conditions: $x = 9,\ 3 \le y \le 6;\ \frac{\partial T}{\partial n} = \lambda_3^I$

$x = 18,\ 3 \le y \le 6;\ T = 0$

$y = 3,\ 9 \le x \le 18;\ \frac{\partial T}{\partial n} = \lambda_2^I$

$y = 6,\ 9 \le x \le 18;\ \frac{\partial T}{\partial n} = 0$

At the I-th iteration between successive time steps, let $T_{11}^I$ and $T_{14}^I$, $T_{21}^I$ and $T_{22}^I$, $T_{34}^I$ and $T_{33}^I$, and $T_{43}^I$ and $T_{42}^I$ be the temperature values computed at interfaces from subdomains 1, 2, 3 and 4 respectively. Then the guessed gradient boundary condition $\lambda$ is updated using Uzawa's algorithm as follows.

$$\lambda_1^{I+1} = \lambda_1^I + \rho\ (T_{21}^I - T_{11}^I)\ /\ L \qquad (2.18)$$
$$\lambda_2^{I+1} = \lambda_2^I + \rho\ (T_{42}^I - T_{22}^I)\ /\ L$$
$$\lambda_3^{I+1} = \lambda_3^I + \rho\ (T_{43}^I - T_{33}^I)\ /\ L$$
$$\lambda_4^{I+1} = \lambda_4^I + \rho\ (T_{34}^I - T_{14}^I)\ /\ L$$

As stated earlier, values of $L = 1$ and $\rho = 0.2$ have been

used in the present work. The flow chart for implementing Uzawa's algorithm is similar to the one given in the context of 2 subdomains.

### 2.4.4 Movement of a Front: 9 - Subdomains

Governing differential equation: $T_t = \frac{1}{Pe} \nabla^2 T$

$$0 < x < 18, \quad 0 < y < 6, \quad t > 0 \tag{2.19}$$

Initial condition: $T = 0$ at $t = 0$

Boundary conditions: $x = 0, \ 0 \le y \le 6; \ T = 1$

$x = 18, \ 0 \le y \le 6; \ T = 0$

$y = 0$ and $6, \ 0 \le x \le 18; \ \frac{\partial T}{\partial n} = 0$

Figure (2.6) shows the decomposition of original problem into 9 equal subdomains. The mathematical problem solved in each subdomain are given below.

Subdomain 1

Governing differential equation: $T_t = \frac{1}{Pe} \nabla^2 T$

$$0 < x < 6, \quad 0 < y < 2, \quad t > 0 \tag{2.20}$$

Initial condition: $T = 0$ at $t = 0$

Boundary conditions: $x = 0, \ 0 \le y \le 2; \ T = 1$

$x = 6, \ 0 \le y \le 2; \ \frac{\partial T}{\partial n} = -\lambda_1^I$

$y = 0, \ 0 \le x \le 6; \ \frac{\partial T}{\partial n} = 0$

$y = 2, \ 0 \le x \le 6; \ \frac{\partial T}{\partial n} = -\lambda_2^I$

Subdomain 2

Governing differential equation: $T_t = \frac{1}{Pe} \nabla^2 T$

$$0 < x < 6, \quad 2 < y < 4, \quad t > 0 \tag{2.21}$$

Initial condition: $T = 0$ at $t = 0$

Boundary conditions: $x = 0, \ 2 \le y \le 4; \ T = 1$

$x = 6, \ 2 \le y \le 4; \ \frac{\partial T}{\partial n} = -\lambda_3^I$

$y = 2, \ 0 \le x \le 6; \ \frac{\partial T}{\partial n} = \lambda_2^I$

$y = 4, \ 0 \le x \le 6; \ \frac{\partial T}{\partial n} = -\lambda_4^I$

Subdomain 3

Governing differential equation: $T_t = \frac{1}{Pe} \nabla^2 T$

$0 < x < 6, \quad 4 < y < 6, \quad t > 0$ (2.22)

Initial condition: $T = 0$ at $t = 0$

Boundary conditions: $x = 0, \ 4 \le y \le 6; \ T = 1$

$x = 6, \ 4 \le y \le 6; \ \frac{\partial T}{\partial n} = -\lambda_5^I$

$y = 4, \ 0 \le x \le 6; \ \frac{\partial T}{\partial n} = \lambda_4^I$

$y = 6, \ 0 \le x \le 6; \ \frac{\partial T}{\partial n} = 0$

Subdomain 4

Governing differential equation: $T_t = \frac{1}{Pe} \nabla^2 T$

$6 < x < 12, \ 4 < y < 6, \quad t > 0$ (2.23)

Initial condition: $T = 0$ at $t = 0$

Boundary conditions: $x = 6, \ 4 \le y \le 6; \ \frac{\partial T}{\partial n} = \lambda_5^I$

$x = 12, \ 4 \le y \le 6; \ \frac{\partial T}{\partial n} = -\lambda_7^I$

$y = 4, \ 6 \le x \le 12; \ \frac{\partial T}{\partial n} = \lambda_6^I$

$y = 6, \ 6 \le x \le 12; \ \frac{\partial T}{\partial n} = 0$

Subdomain 5

Governing differential equation: $T_t = \frac{1}{Pe} \nabla^2 T$

$12 < x < 18, \quad 4 < y < 6, \quad t > 0$ (2.24)

Initial condition: $T = 0$ at $t = 0$

Boundary conditions: $x = 12, \ 4 \le y \le 6; \ \frac{\partial T}{\partial n} = \lambda_7^I$

$x = 18, \ 4 \le y \le 6; \ T = 0$

$y = 4, \ 12 \le x \le 18; \ \frac{\partial T}{\partial n} = \lambda_8^I$

$y = 6, \ 12 \le x \le 18; \ \frac{\partial T}{\partial n} = 0$

Subdomain 6

Governing differential equation: $T_t = \frac{1}{Pe} \nabla^2 T$

$12 < x < 18, \quad 2 < y < 4, \quad t > 0$ (2.25)

Initial condition: $T = 0$ at $t = 0$

Boundary conditions: $x = 12,\ 2 \le y \le 4;\ \frac{\partial T}{\partial n} = \lambda_9^I$

$x = 18,\ 2 \le y \le 4;\ T = 0$

$y = 2,\ 12 \le x \le 18;\ \frac{\partial T}{\partial n} = \lambda_{10}^I$

$y = 4,\ 12 \le x \le 18;\ \frac{\partial T}{\partial n} = -\lambda_8^I$

Subdomain 7

Governing differential equation: $T_t = \frac{1}{Pe} \nabla^2 T$

$12 < x < 18,\quad 0 < y < 2,\quad t > 0$ (2.26)

Initial condition: $T = 0$ at $t = 0$

Boundary conditions: $x = 12,\ 0 \le y \le 2;\ \frac{\partial T}{\partial n} = \lambda_{11}^I$

$x = 18,\ 0 \le y \le 2;\ T = 0$

$y = 0,\ 12 \le x \le 18;\ \frac{\partial T}{\partial n} = 0$

$y = 2,\ 12 \le x \le 18;\ \frac{\partial T}{\partial n} = -\lambda_{10}^I$

Subdomain 8

Governing differential equation: $T_t = \frac{1}{Pe} \nabla^2 T$

$6 < x < 12,\quad 0 < y < 2,\quad t > 0$ (2.27)

Initial condition: $T = 0$ at $t = 0$

Boundary conditions: $x = 6,\ 0 \le y \le 2;\ \frac{\partial T}{\partial n} = \lambda_1^I$

$x = 12,\ 0 \le y \le 2;\ \frac{\partial T}{\partial n} = -\lambda_{11}^I$

$y = 0,\ 6 \le x \le 12;\ \frac{\partial T}{\partial n} = 0$

$y = 2,\ 6 \le x \le 12;\ \frac{\partial T}{\partial n} = -\lambda_{12}^I$

Subdomain 9

Governing differential equation: $T_t = \frac{1}{Pe} \nabla^2 T$

$6 < x < 12,\quad 2 < y < 4,\quad t > 0$ (2.28)

Initial condition: $T = 0$ at $t = 0$

Boundary conditions: $x = 6,\ 2 \le y \le 4;\ \frac{\partial T}{\partial n} = \lambda_3^I$

$x = 12,\ 2 \le y \le 4;\ \frac{\partial T}{\partial n} = -\lambda_9^I$

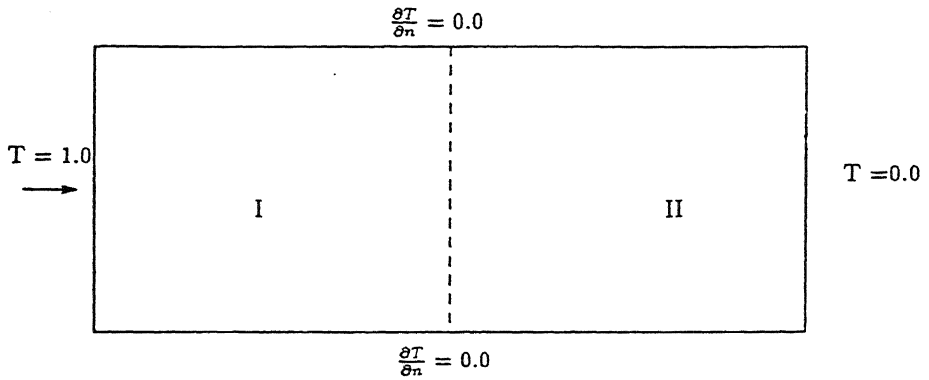



Figure 2.3 Division of the physical domain into Two Subdomains

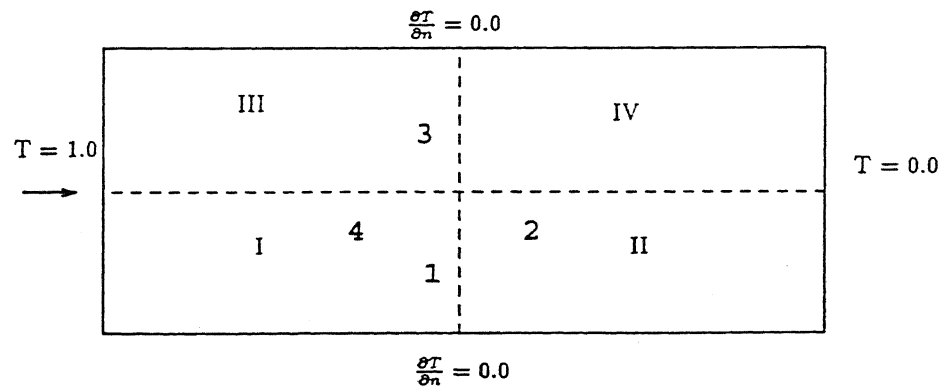



Figure 2.5 Division of the Computational Domain into Four Subdomains

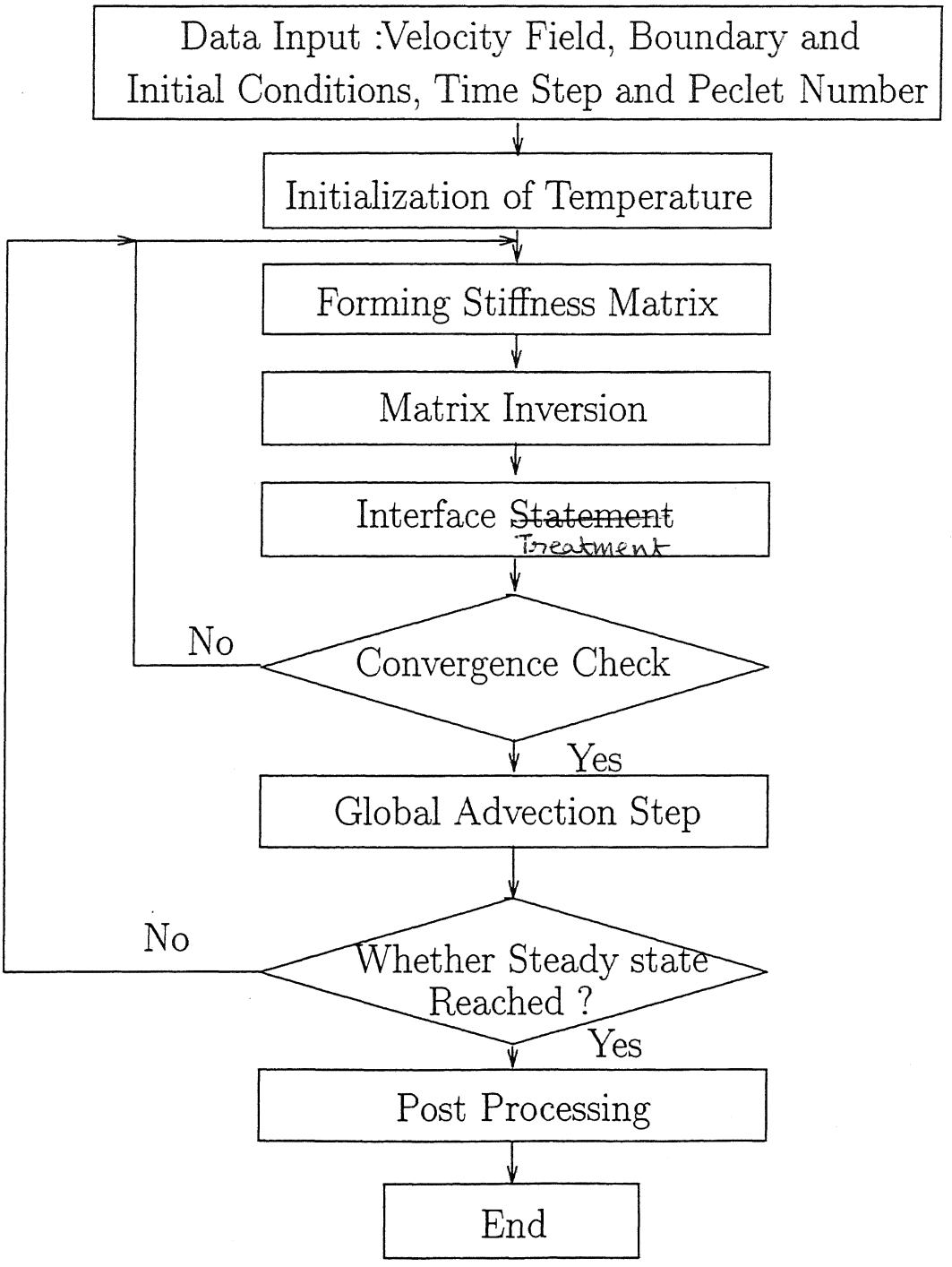



Figure 2.4 Flow chart for Domain Decomposition with Uzawa' Algorithm

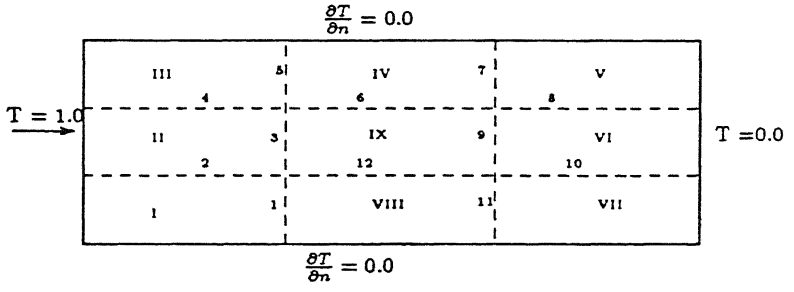



Figure 2.6 Division of the Computational Domain into Nine Subdomains

$$y = 2,\ 6 \le x \le 12;\ \frac{\partial T}{\partial n} = \lambda_{12}^{I}$$

$$y = 4,\ 6 \le x \le 12;\ \frac{\partial T}{\partial n} = -\lambda_{6}^{I}$$

At the I-th iteration between successive time steps, let $T_{11}^{I}$ and $T_{12}^{I}$, $T_{22}^{I}$, $T_{23}^{I}$ and $T_{24}^{I}$, $T_{34}^{I}$ and $T_{35}^{I}$, $T_{45}^{I}$, $T_{46}^{I}$ and $T_{47}^{I}$, $T_{57}^{I}$ and $T_{58}^{I}$, $T_{68}^{I}$, $T_{69}^{I}$ and $T_{610}^{I}$, $T_{710}^{I}$ and $T_{711}^{I}$, $T_{811}^{I}$, $T_{812}^{I}$ and $T_{81}^{I}$, $T_{93}^{I}$, $T_{96}^{I}$, $T_{99}^{I}$ and $T_{912}^{I}$ be temperature values computed at the interfaces from subdomains 1, 2, 3, 4, 5, 6, 7, 8 and 9 respectively. In the notation $T_{xy}^{I}$, x represents subdomain number, y represents interface number and I represents iteration number. The guessed gradient boundary condition $\lambda$ is updated using Uzawa's algorithm as follows

$$\lambda_{1}^{I+1} = \lambda_{1}^{I} + \rho\ (T_{81}^{I} - T_{11}^{I})\ /\ L \qquad (2.29)$$

$$\lambda_{2}^{I+1} = \lambda_{2}^{I} + \rho\ (T_{22}^{I} - T_{12}^{I})\ /\ L$$

$$\lambda_{3}^{I+1} = \lambda_{3}^{I} + \rho\ (T_{93}^{I} - T_{23}^{I})\ /\ L$$

$$\lambda_{4}^{I+1} = \lambda_{4}^{I} + \rho\ (T_{34}^{I} - T_{24}^{I})\ /\ L$$

$$\lambda_{5}^{I+1} = \lambda_{5}^{I} + \rho\ (T_{45}^{I} - T_{35}^{I})\ /\ L$$

$$\lambda_{6}^{I+1} = \lambda_{6}^{I} + \rho\ (T_{46}^{I} - T_{96}^{I})\ /\ L$$

$$\lambda_{7}^{I+1} = \lambda_{7}^{I} + \rho\ (T_{57}^{I} - T_{47}^{I})\ /\ L$$

$$\lambda_{8}^{I+1} = \lambda_{8}^{I} + \rho\ (T_{58}^{I} - T_{68}^{I})\ /\ L$$

$$\lambda_{9}^{I+1} = \lambda_{9}^{I} + \rho\ (T_{69}^{I} - T_{99}^{I})\ /\ L$$

$$\lambda_{10}^{I+1} = \lambda_{10}^{I} + \rho\ (T_{610}^{I} - T_{710}^{I})\ /\ L$$

$$\lambda_{11}^{I+1} = \lambda_{11}^{I} + \rho\ (T_{711}^{I} - T_{811}^{I})\ /\ L$$

$$\lambda_{12}^{I+1} = \lambda_{12}^{I} + \rho\ (T_{912}^{I} - T_{812}^{I})\ /\ L$$

### 2.4.5 Heat Transfer from a Single Cylinder Using 2-Subdomains

Governing differential equation: $T_t = \frac{1}{Pe} \nabla^2 T$

$$0 < x < 12,\quad 0 < y < 6,\quad t > 0 \qquad (2.30)$$

Initial condition: $T = 0$ at $t = 0$

Boundary conditions: $x = 0,\ 0 \le y \le 6;\ T = 0$

$$x = 12,\ 0 \le y \le 6;\ \frac{\partial T}{\partial n} = 0$$

$$y = 0 \text{ and } 6,\ 0 \le x \le 12;\ \frac{\partial T}{\partial n} = 0$$

On the cylinder surface, we consider a constant heat flux boundary condition

$$\frac{\partial T}{\partial n} = -1,$$

where, n is a unit outward drawn normal on the cylinder surface.

Figure (2.7) shows the decomposition of original problem into 2 equal subdomains. Each subdomain has a size of 6 x 6 units. The mathematical problem solved in each subdomain are given below.

Subdomain 1

Governing differential equation: $T_t = \frac{1}{Pe} \nabla^2 T,$

$$0 < x < 6,\quad 0 < y < 6,\quad t > 0 \qquad (2.31)$$

Initial condition: $T = 0$ at $t = 0$

Boundary conditions: $x = 0,\ 0 \le y \le 6;\ T = 0$

$$x = 6,\ 0 \le y \le 6;\ \frac{\partial T}{\partial n} = -\lambda^I$$

$$y = 0 \text{ and } 6,\ 0 \le x \le 6;\ \frac{\partial T}{\partial n} = 0$$

On the cylinder surface, $-\frac{\partial T}{\partial n} = 1.$

Subdomain 2

Governing differential equation: $T_t = \frac{1}{Pe} \nabla^2 T$

$$6 < x < 12,\quad 0 < y < 6,\quad t > 0 \qquad (2.32)$$

Initial condition: $T = 0$ at $t = 0$

Boundary conditions: $x = 6,\ 0 \le y \le 6;\ \frac{\partial T}{\partial n} = \lambda^I$

$$x = 12,\ 0 \le y \le 6;\ \frac{\partial T}{\partial n} = 0$$

$$y = 0 \text{ and } 6,\ 6 \le x \le 12;\ \frac{\partial T}{\partial n} = 0$$

Equations (2.31) and (2.32) are solved using Bubnov-Galerkin finite element method (See Appendix A).

At the I-th iteration between successive time steps let $T_1^I$ and $T_2^I$ be the temperature values computed at the interface from subdomains 1 and 2 respectively. Then the guessed gradient boundary condition $\lambda$ is updated using Uzawa's algorithm as follows.

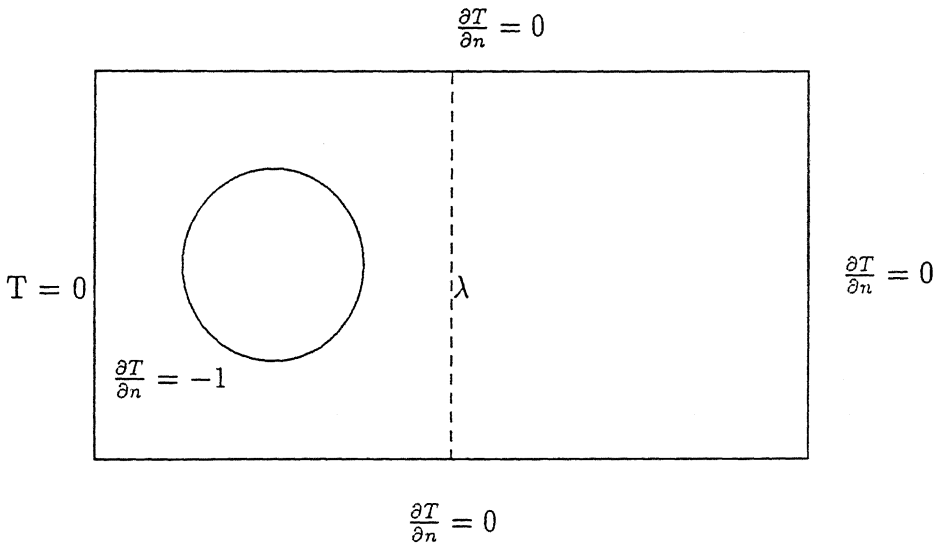



Figure 2.7 Division of the Computational Domain with Single Cylinder into Two Subdomains

$$\lambda^{I+1} = \lambda^{I} + \rho\ (T_2^I - T_1^I)\ /\ L \qquad (2.33)$$

As before values of L = 1 and $\rho$ = 0.2 have been used in this study. Figure 2.4 shows a flow chart for implementing Uzawa's algorithm.

### 2.4.6 Single Cylinder With 9-Subdomains

The full domain is divided into 9-subdomains with one of them containing the circular cylinder. The central subdomain contains the cylinder. All other subdomains are simple rectangles.

The governing differential equation, initial conditions, boundary conditions are given below for the full domain.

Governing differential equation: $T_t = \frac{1}{Pe} \nabla^2 T$

$$0 < x < 6, \quad 0 < y < 6, \quad t > 0 \qquad (2.34)$$

Initial condition: $T = 0$ at $t = 0$

Boundary conditions: $x = 0,\ 0 \le y \le 6;\ T = 0$

$x = 6,\ 0 \le y \le 6;\ \frac{\partial T}{\partial n} = 0$

$y = 0$ and $6,\ 0 \le x \le 6;\ \frac{\partial T}{\partial n} = 0$

On the cylinder surface, we consider a constant heat flux boundary condition

$$\frac{\partial T}{\partial n} = -1,$$

where, n is a unit outward drawn normal on the cylinder surface.

The mathematical problem solved in each subdomain is given below.

Subdomain 1

Governing differential equation: $T_t = \frac{1}{Pe} \nabla^2 T$

$$0 < x < 1, \quad 0 < y < 1, \quad t > 0 \qquad (2.35)$$

Initial condition: $T = 0$ at $t = 0$

Boundary conditions: $x = 0,\ 0 \le y \le 1;\ T = 0$

$x = 1,\ 0 \le y \le 1;\ \frac{\partial T}{\partial n} = \lambda_1^I$

$y = 0,\ 0 \le x \le 1;\ \frac{\partial T}{\partial n} = 0$

$y = 1,\ 0 \le x \le 1;\ \frac{\partial T}{\partial n} = \lambda_2^I$

Subdomain 2

Governing differential equation: $T_t = \frac{1}{Pe} \nabla^2 T$

$0 < x < 1, \quad 1 < y < 5, \quad t > 0$ (2.36)

Initial condition: $T = 0$ at $t = 0$

Boundary conditions: $x = 0, \ 1 \le y \le 5; \ T = 0$

$x = 1, \ 1 \le y \le 5; \ \frac{\partial T}{\partial n} = \lambda_3^I$

$y = 1, \ 0 \le x \le 1; \ \frac{\partial T}{\partial n} = - \lambda_2^I$

$y = 5, \ 0 \le x \le 1; \ \frac{\partial T}{\partial n} = - \lambda_4^I$

Subdomain 3

Governing differential equation: $T_t = \frac{1}{Pe} \nabla^2 T$

$0 < x < 1, \quad 5 < y < 6, \quad t > 0$ (2.37)

Initial condition: $T = 0$ at $t = 0$

Boundary conditions: $x = 0, \ 5 \le y \le 6; \ T = 0$

$x = 1, \ 5 \le y \le 6; \ \frac{\partial T}{\partial n} = \lambda_5^I$

$y = 5, \ 0 \le x \le 1; \ \frac{\partial T}{\partial n} = \lambda_4^I$

$y = 6, \ 0 \le x \le 1; \ \frac{\partial T}{\partial n} = 0$

Subdomain 4

Governing differential equation: $T_t = \frac{1}{Pe} \nabla^2 T$

$1 < x < 5, \quad 5 < y < 6, \quad t > 0$ (2.38)

Initial condition: $T = 0$ at $t = 0$

Boundary conditions: $x = 1, \ 5 \le y \le 6; \ \frac{\partial T}{\partial n} = - \lambda_5^I$

$x = 5, \ 5 \le y \le 6; \ \frac{\partial T}{\partial n} = - \lambda_7^I$

$y = 5, \ 1 \le x \le 5; \ \frac{\partial T}{\partial n} = \lambda_6^I$

$y = 6, \ 1 \le x \le 5; \ \frac{\partial T}{\partial n} = 0$

Subdomain 5

Governing differential equation: $T_t = \frac{1}{Pe} \nabla^2 T$

$5 < x < 6, \quad 5 < y < 6, \quad t > 0$ (2.39)

Initial condition: $T = 0$ at $t = 0$

Boundary conditions: $x = 5,\ 5 \le y \le 6;\ \frac{\partial T}{\partial n} = \lambda_7^I$

$x = 6,\ 5 \le y \le 6;\ \frac{\partial T}{\partial n} = 0$

$y = 5,\ 5 \le x \le 6;\ \frac{\partial T}{\partial n} = \lambda_8^I$

$y = 6,\ 5 \le x \le 6;\ \frac{\partial T}{\partial n} = 0$

Subdomain 6

Governing differential equation: $T_t = \frac{1}{Pe} \nabla^2 T$

$5 < x < 6,\quad 1 < y < 5,\quad t > 0$ (2.40)

Initial condition: $T = 0$ at $t = 0$

Boundary conditions: $x = 5,\ 1 \le y \le 5;\ \frac{\partial T}{\partial n} = \lambda_9^I$

$x = 6,\ 1 \le y \le 5;\ \frac{\partial T}{\partial n} = 0$

$y = 1,\ 5 \le x \le 6;\ \frac{\partial T}{\partial n} = -\lambda_{10}^I$

$y = 5,\ 5 \le x \le 6;\ \frac{\partial T}{\partial n} = -\lambda_8^I$

Subdomain 7

Governing differential equation: $T_t = \frac{1}{Pe} \nabla^2 T$

$5 < x < 6,\quad 0 < y < 1,\quad t > 0$ (2.41)

Initial condition: $T = 0$ at $t = 0$

Boundary conditions: $x = 5,\ 0 \le y \le 1;\ \frac{\partial T}{\partial n} = \lambda_{11}^I$

$x = 6,\ 0 \le y \le 1;\ \frac{\partial T}{\partial n} = 0$

$y = 0,\ 5 \le x \le 6;\ \frac{\partial T}{\partial n} = 0$

$y = 1,\ 5 \le x \le 6;\ \frac{\partial T}{\partial n} = \lambda_{10}^I$

Subdomain 8

Governing differential equation: $T_t = \frac{1}{Pe} \nabla^2 T$

$1 < x < 5,\quad 0 < y < 1,\quad t > 0$ (2.42)

Initial condition: $T = 0$ at $t = 0$

Boundary conditions: $x = 1,\ 0 \le y \le 1;\ \frac{\partial T}{\partial n} = -\lambda_1^I$

$x = 5,\ 0 \le y \le 1;\ \frac{\partial T}{\partial n} = -\lambda_{11}^I$

$$y = 0,\ 1 \le x \le 5;\ \frac{\partial T}{\partial n} = 0$$
$$y = 1,\ 1 \le x \le 5;\ \frac{\partial T}{\partial n} = \lambda_{12}^{I}$$

Subdomain 9

Governing differential equation: $T_t = \frac{1}{Pe} \nabla^2 T$

$$1 < x < 5,\quad 1 < y < 5,\quad t > 0 \qquad (2.43)$$

Initial condition: $T = 0$ at $t = 0$

Boundary conditions:
$$x = 1,\ 1 \le y \le 5;\ \frac{\partial T}{\partial n} = -\lambda_{3}^{I}$$
$$x = 5,\ 1 \le y \le 5;\ \frac{\partial T}{\partial n} = -\lambda_{9}^{I}$$
$$y = 1,\ 1 \le x \le 5;\ \frac{\partial T}{\partial n} = -\lambda_{12}^{I}$$
$$y = 5,\ 1 \le x \le 5;\ \frac{\partial T}{\partial n} = -\lambda_{6}^{I}$$

On the cylinder surface, $-\frac{\partial T}{\partial n} = 1$.

Equations (2.35) to (2.43) are solved using Galerkin finite element method. At the I-th iteration between successive time steps, let $T_1^I$ and $T_{12}^I$, $T_{22}^I$, $T_{23}^I$ and $T_{24}^I$, $T_{34}^I$ and $T_{35}^I$, $T_{45}^I$, $T_{46}^I$ and $T_{47}^I$, $T_{57}^I$ and $T_{58}^I$, $T_{68}^I$, $T_{69}^I$ and $T_{610}^I$, $T_{710}^I$ and $T_{711}^I$, $T_{811}^I$, $T_{812}^I$ and $T_{81}^I$, $T_{93}^I$, $T_{96}^I$, $T_{99}^I$ and $T_{912}^I$ be temperature values computed at the interfaces from subdomains 1, 2, 3, 4, 5, 6, 7, 8 and 9 respectively. In the notation $T_{xy}^I$, x represents subdomain number, y represents interface number and I represents iteration number. The guessed gradient boundary condition $\lambda$ is updated using Uzawa's algorithm as follows

$$\lambda_1^{I+1} = \lambda_1^I + \rho\ (T_{11}^I - T_{81}^I)\ /\ L \qquad (2.44)$$
$$\lambda_2^{I+1} = \lambda_2^I + \rho\ (T_{12}^I - T_{22}^I)\ /\ L$$
$$\lambda_3^{I+1} = \lambda_3^I + \rho\ (T_{23}^I - T_{93}^I)\ /\ L$$
$$\lambda_4^{I+1} = \lambda_4^I + \rho\ (T_{34}^I - T_{24}^I)\ /\ L$$

$$\lambda_5^{I+1} = \lambda_5^I + \rho\ (T_{35}^I - T_{45}^I)\ /\ L$$
$$\lambda_6^{I+1} = \lambda_6^I + \rho\ (T_{46}^I - T_{96}^I)\ /\ L$$
$$\lambda_7^{I+1} = \lambda_7^I + \rho\ (T_{57}^I - T_{47}^I)\ /\ L$$
$$\lambda_8^{I+1} = \lambda_8^I + \rho\ (T_{58}^I - T_{68}^I)\ /\ L$$
$$\lambda_9^{I+1} = \lambda_9^I + \rho\ (T_{69}^I - T_{99}^I)\ /\ L$$
$$\lambda_{10}^{I+1} = \lambda_{10}^I + \rho\ (T_{710}^I - T_{610}^I)\ /\ L$$
$$\lambda_{11}^{I+1} = \lambda_{11}^I + \rho\ (T_{711}^I - T_{811}^I)\ /\ L$$
$$\lambda_{12}^{I+1} = \lambda_{12}^I + \rho\ (T_{812}^I - T_{912}^I)\ /\ L$$

## 2.5 VELOCITY FIELD FOR 1 AND 5 CYLINDERS

In the front problem, the velocity field is prescribed as u = 1 and v = 0, a constant. For flow past an array of cylinders, the flow is assumed to be Darcian, i.e., flow is assumed to be governed by Darcy's law.

$$u = -\frac{k}{\mu}\nabla P \qquad (2.45)$$

The fluid is taken to be incompressible and so $\nabla.u = 0$. The boundary condition u.n = 0 is applied on all impermeable surfaces. For flow past a single cylinder these equations can be solved using potential theory. This yields,

$$u - i\,v = 1 - \frac{1}{z^2}\ ,\ \text{where } z = x + i\,y,\quad i = \sqrt{-1}$$

and u and v are cartesian components of velocity. For flow past an array of cylinders the permeability k appearing in Darcy law is made a spatial variable with a value of unity in the fluid region and a small value, around $10^{-7}$ over the area occupied by the cylinder. The velocity problem is then solved numerically in terms of a stream function $\psi$ which is related to velocity as $u = (\psi_y, -\psi_x)$. the equation governing $\psi$ can be shown to be

$$\nabla.\frac{1}{k}\nabla\psi = 0 \qquad (2.46)$$

with the approach flow condition $(\psi_y, -\psi_x) = (1, 0)$ i.e. $\psi = y$ ahead of the cylinder array.

In the present study calculations have been carried out on a square domain in which a single cylinder or array of cylinders is present. This is shown in Figure (2.8). With reference to this

figure the boundary conditions for Equation (2.46) are

$$x = 0,\ \psi = y;\ y = 0,\ \psi = 0$$

$$y = L,\ \psi = L \text{ and } x = L,\ \psi_x = 0 \qquad (2.47)$$

Equation (2.46) has been solved by a control volume finite difference method with harmonic averaging for k at the interface between the cylinder and the porous region. The grid size of 101 x 101 has been found to adequate to obtain grid independent velocity profiles. Streamline pattern is shown in Figure (2.9).

## 2.6 GRID GENERATION

Partial differential equation based methods are commonly used for structured grid generation in a wide variety of applications. These techniques may be termed as elliptic, parabolic or hyperbolic, depending upon the characteristics of the grid generation equations. In most applications the grid generation equation are also transformed on the rectangular computational domain and solved along with the governing equations of the physical problem. Elliptic grid generation methods offer good grid smoothness and spacing control, and have been employed in this work. Elements generated by grid generation are quadrilaterals. These have been subsequently subdivided into triangles for finite element calculations.

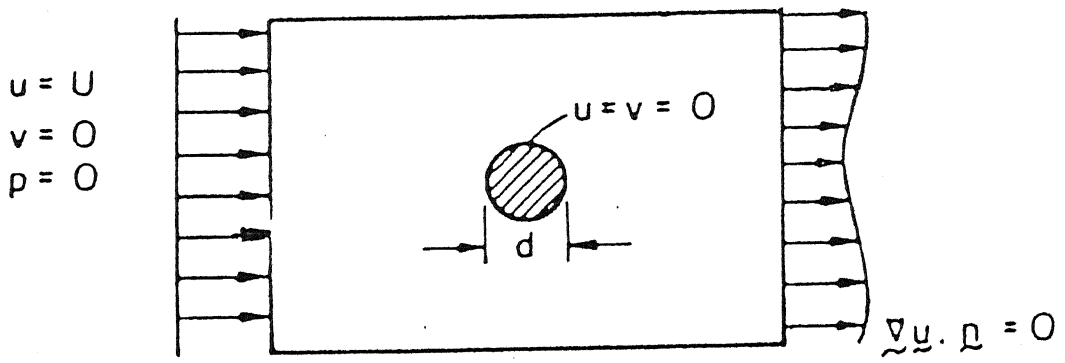



Figure 2.8 Schematic Diagram of Computational Domain for Flow Field

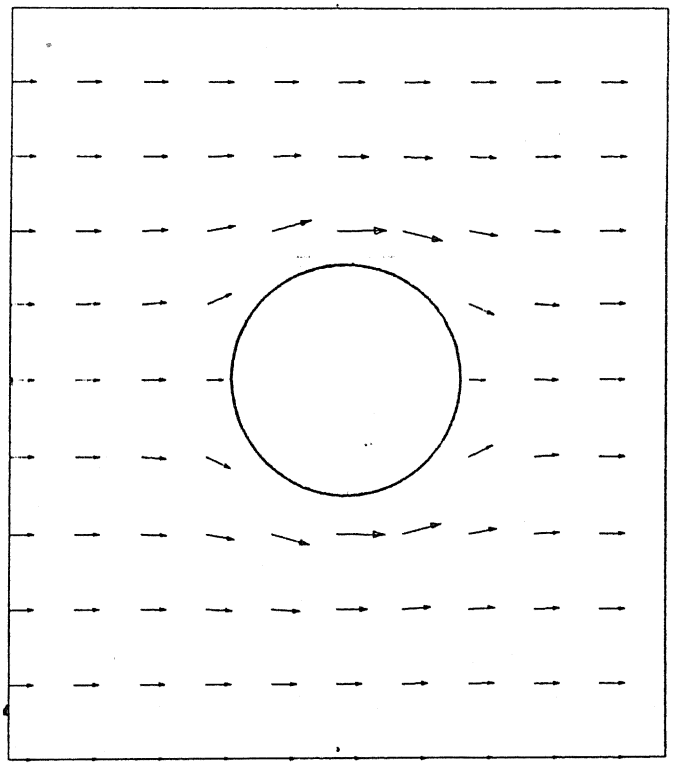


Figure 2.9 Streamline Pattern in the Computational Domain

# CHAPTER 3

# TESTING OF COMPUTER PROGRAM

## 3.1 NUMERICAL DETAILS

In the present study, a variety of grids have been used depending on whether the problem studied was front movement or flow past a cylinder and the Peclet number. In the absence of cylinders, the grid used is uniform and identical six-noded triangular elements are used. Figure 3.1 shows the grid used in this present study with 216 six-noded triangular elements for problems with Pe ≤ 100. The number of nodes used is 481. The region size considered here is 18 x 6, dimensionless units. For a simulation with one cylinder the grid used is necessarily non-uniform (See Figure 3.2). For a region size of 6 x 6, 256 elements and 576 nodes with 32 nodes on cylinder surface are used for Pe ≤ 10 and for Pe ≥ 50 region size used is same, 576 elements and 1248 nodes with 48 nodes on the cylinder surface are used. Owing to the use of isoparametric elements the representation of the cylinder is exact.

The operator splitting technique is explicit in time. The maximum permissible time step is found by using the courant number restriction, C ≤ 1. In the worst case C = 1 where

$$C = \max \left( \frac{U\ \Delta t_{max}}{\Delta x} , \frac{V\ \Delta t_{max}}{\Delta y} \right) \qquad (3.1)$$

Here the velocities u, v at all nodes and Δx, Δy for all the elements are known before hand. Applying Equation (3.1) to all the nodes, the maximum permissible Δt over the whole mesh can be found. Since the definition of C in Equation (3.1) is only approximate, a value of 50% of that maximum permissible time step has been used for the present calculations. For the front movement problem, Δt = 0.5 for all Peclet numbers. For flow past a single cylinder, Δt = 0.095 for Pe ≤ 10 and Δt = 0.05 for Pe > 10 for the grid shown in Figure 3.2.

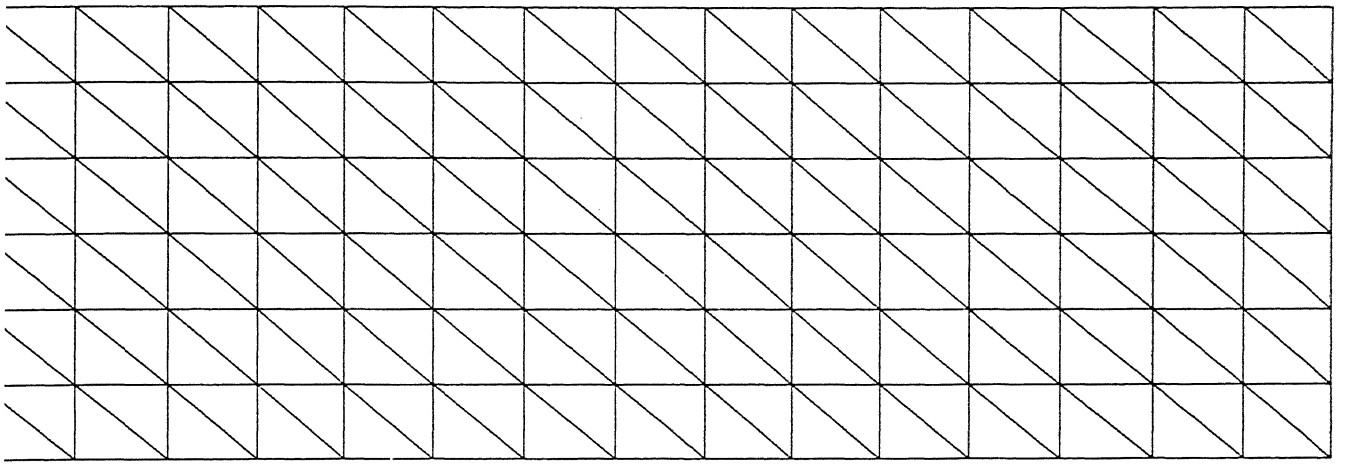


Figure 3.1 Finite Element Mesh of the Computational Domain without Cylinder

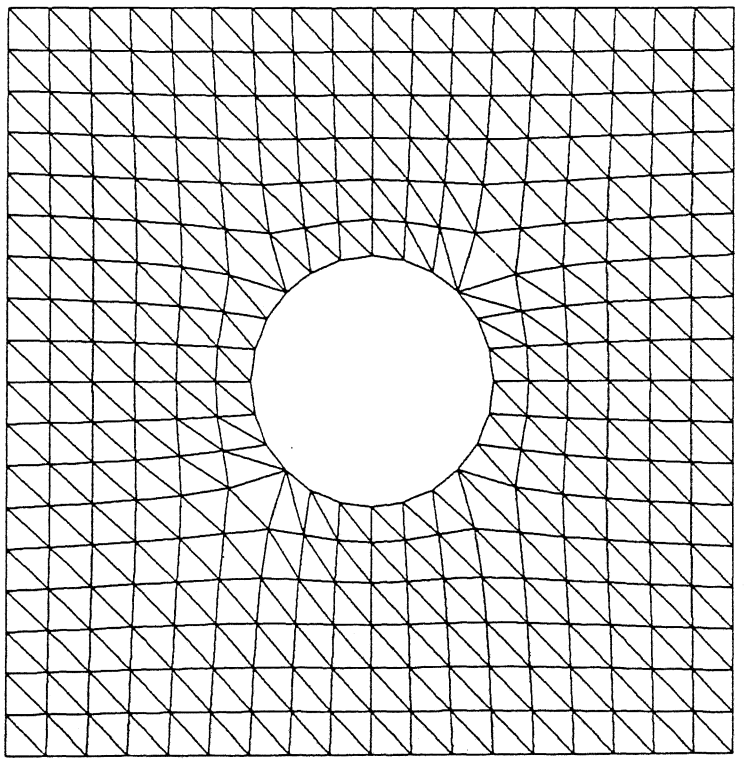


Figure 3.2 Finite Element Mesh of the Computational Domain with Single Cylinder

## 3.2 FRONT MOVEMENT PROBLEM

In this section, we consider steady flow between two parallel insulated surfaces and a step increase in temperature at the inflow plane. The resulting problem is one-dimensional. Operator splitting with universal limiters and finite element solutions over the flow domain are compared with the analytical solution of the problem (see Appendix C for the analytical solution).

Fronts will develop at high Peclet numbers and will propagate in steps of time intervals over the grid spacings. At low Peclet numbers the transport is diffusion dominated. Owing to this, clear fronts are not seen in low Peclet number flows.

The governing differential equation for propagation of a thermal front is

$$T_t + u.\nabla T = \frac{1}{Pe} \nabla^2 T \qquad (3.2)$$

along with boundary conditions

$$T = 1, \text{ at } x = 0 \qquad 0 \le y \le L$$

$$T = 0, \text{ at } x = L \qquad 0 \le y \le L$$

$$\frac{\partial T}{\partial n} = 0 \text{ at } y = 0, L \qquad 0 \le x \le L$$

where L is non-dimensional length of domain.

Further, the initial condition is, $T = 0$ at $t = 0$.

For the analytical solution of Equation (3.2), the outflow boundary condition used is, $x \longrightarrow \infty$, $T \longrightarrow 0$. Two time levels $t = 5$ and $t = 10$ are considered for comparing the numerical and analytical solutions.

For $Pe = 0.1$ and at $t = 5$ and $t = 10$ a gradual temperature profile moves through the fluid domain (Figure 3.3). The analytical solution deviates from operator splitting and finite element method after some distance ($x = 11$) because the analytical solution has the different far field boundary condition namely $T \longrightarrow 0$ at $x \longrightarrow \infty$. However, the numerical methods compare closely with one another.

At $Pe = 1.0$ all the 4 methods give identical answers (Figure 3.4). At $Pe = 10$ at both time levels ($t = 5$ and $t = 10$) a sharp front is clearly seen (Figure 3.5). Here operator splitting and analytical methods coincide for all values of x. The conventional

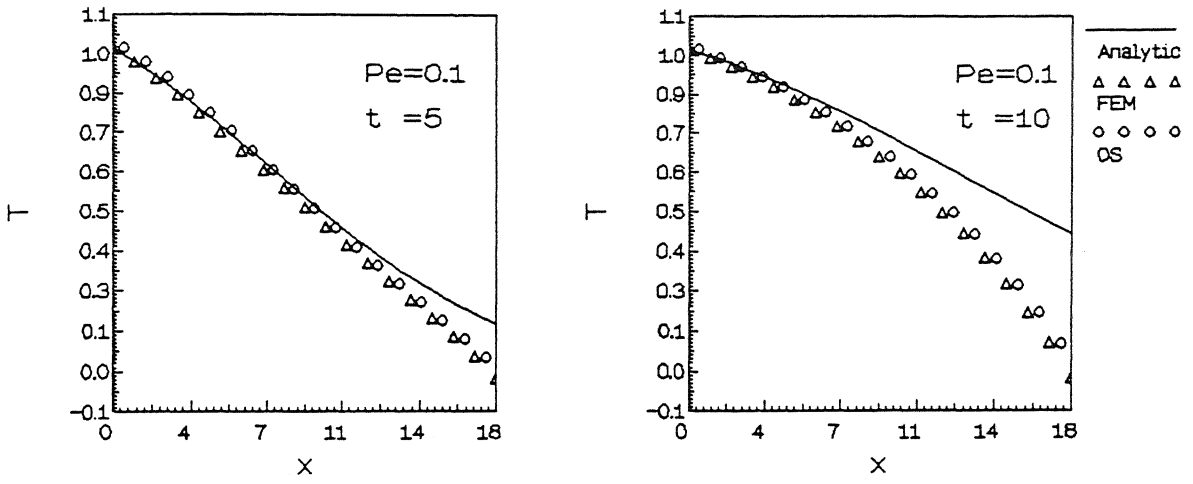



Figure 3.3 Variation of Temperature along Flow Direction for Pe = 0.1

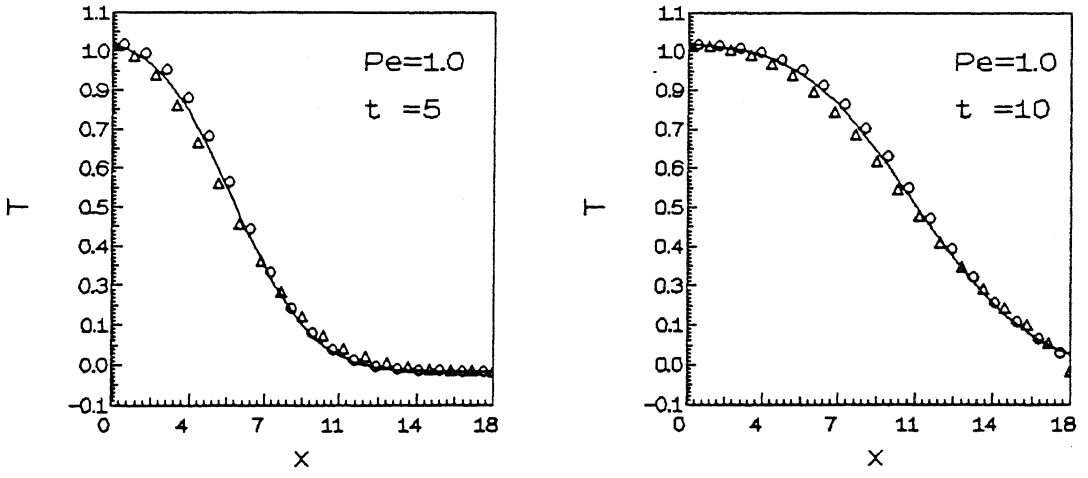



Figure 3.4 Variation of Temperature along Flow Direction for Pe = 1

finite element results deviate at positions of sharp gradient because of round off errors during inversion of the unsymmetric matrix. Hence, the Bubnov-Galerkin method is not directly suitable for highly convective flows.

The same trend as at Pe = 10 is maintained at Pe = 100. The performance of operator splitting is very clear compared to that at Pe = 10. The fronts at Pe = 100 are shown in Figure (3.6). Operator splitting and operator splitting with limiters yield identical answers and the latter has not been explicitly shown.

### 3.3 FLOW PAST A SINGLE CYLINDER: COMPARISON WITH BOUNDARY LAYER METHOD

A heated cylinder placed in Darcian flow will develop a thermal boundary layer over its surface at high flow rates. Boundary layer theory is valid at steady state and for single cylinder problems at high Peclet number flows, it can then be used to determine heat transfer rates from the cylinder. Boundary layer theory is not valid during transients, low Peclet numbers and in cylinder array problems. Boundary layer solution for flow past a cylinder is presented in Appendix D. Using this approach we get local Nusselt number over the cylinder surface and the average Nusselt number.

Figure (3.7) shows steady Nusselt number variation on the cylinder surface at Pe = 0.1, Pe = 1.0 and Pe = 10. The operator splitting and finite element method results are seen to coincide, but deviate from boundary layer theory. At Pe = 10, the boundary layer results coincide with those of operator splitting, but deviate from the finite element results. In this case, finite element method requires a fine mesh for getting ~~of~~ greater accuracy. From this we conclude that operator splitting is efficient even on coarse mesh at high Peclet numbers.

Figures (3.8) shows steady Nusselt number variation on the cylinder surface at Pe = 50 and Pe = 100. The behaviour of operator splitting and operator splitting with universal limiters are somewhat different from boundary layer theory. In particular, the variation of Nusselt number with angle is non-monotonic when numerical methods are used. We can see whether this result is

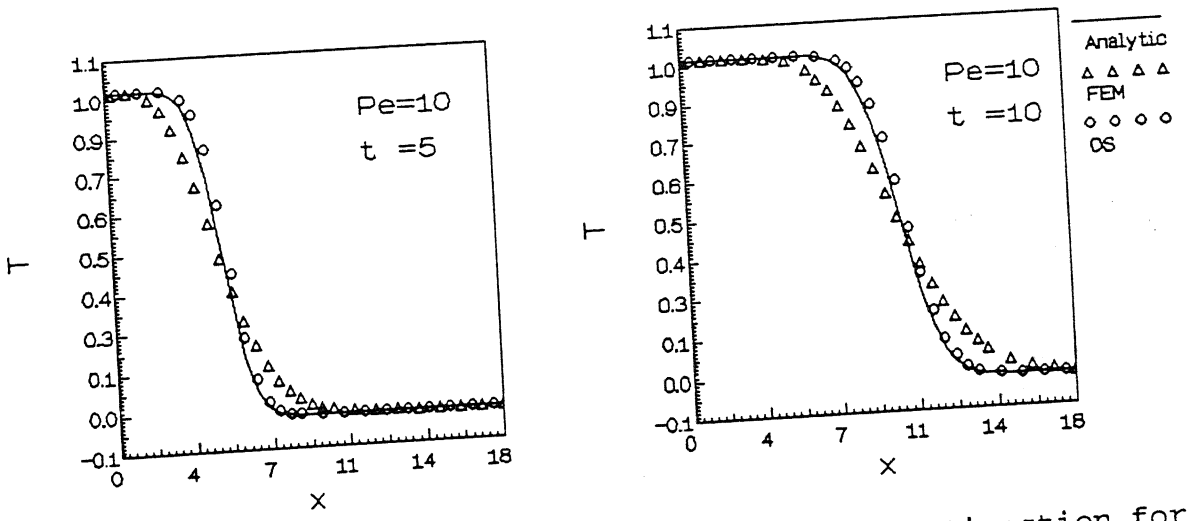



Figure 3.5 Variation of Temperature along Flow Direction for Pe = 10

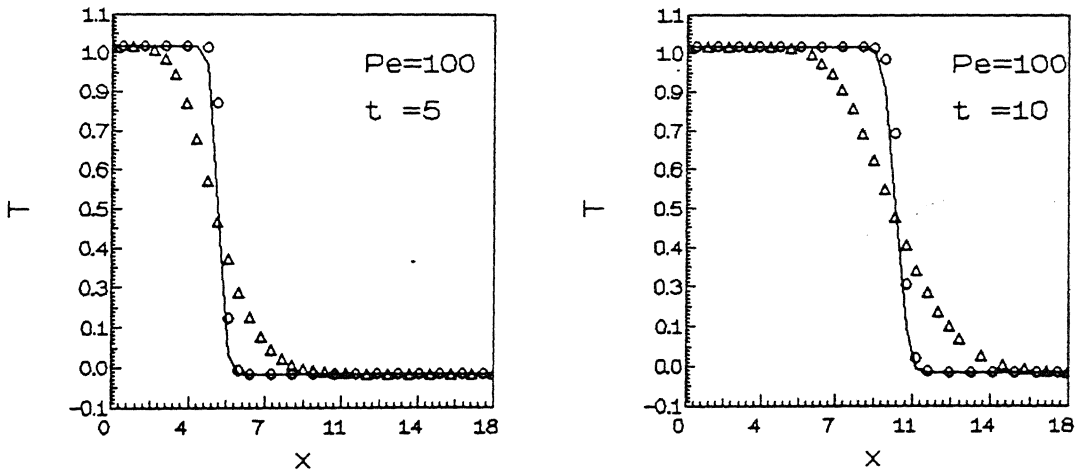



Figure 3.6 Variation of Temperature along Flow Direction for Pe = 100

physical or unphysical by using a fine mesh. This is discussed in Chapter 5 in the context of domain decomposition. Figures (3.7) and (3.8) and Table (3.1) show that operator splitting and operator splitting with limiters are generally accurate methods at high Peclet number flows, compared to the full finite element method (FEM) for a fixed number of elements.

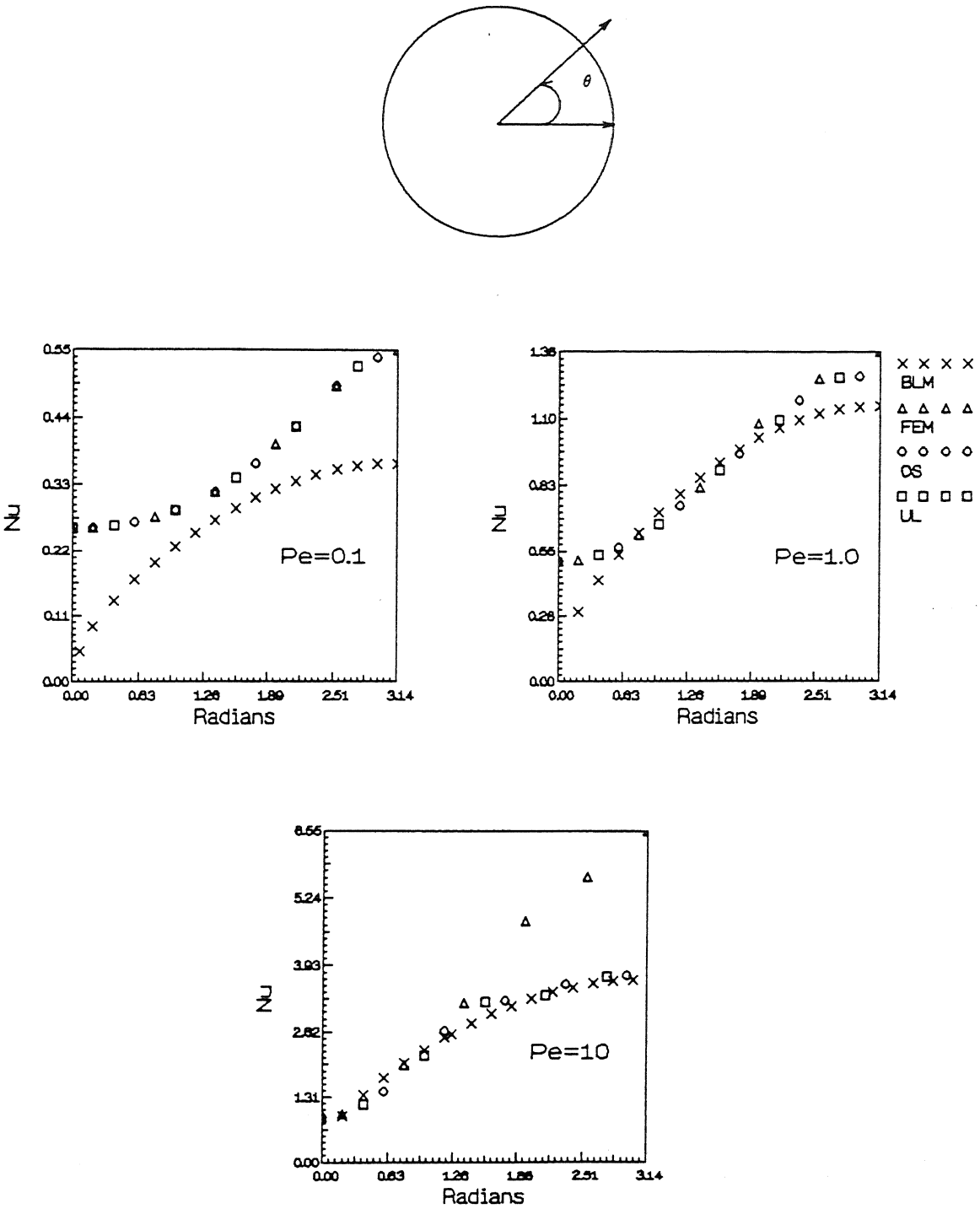



Figure 3.7 Variation of Steady Nusselt Number on the Cylinder Surface for Pe = 0.1,1 and 10

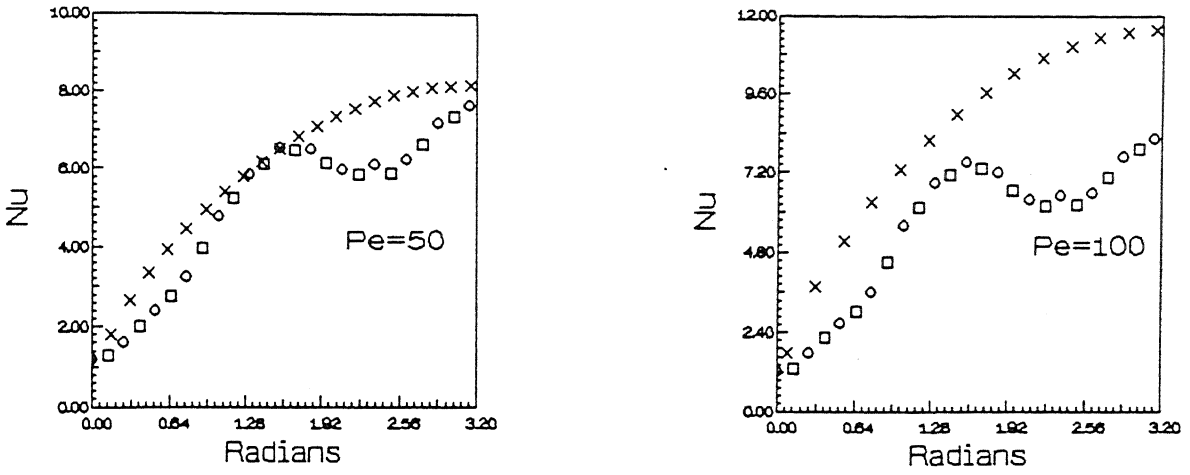



Figure 3.8 Variation of Steady Nusselt Number on the Cylinder Surface for Pe = 50 and 100

Table 3.1 Steady Nusselt Number Values for Different Methods

| Method | Pe = 0.1 | Pe = 1.0 | Pe = 10 | Pe = 100 19 x 19 | Pe = 100 23 x 23 |
|---|---|---|---|---|---|
| BLM | 0.2645 | 0.8368 | 2.646 | 8.368 | 8.368 |
| FEM | 0.370 | 0.922 | 3.762 | 20.071 | 19.844 |
| OS | 0.370 | 0.893 | 2.668 | 5.382 | 5.287 |
| UL | 0.370 | 0.893 | 2.668 | 5.388 | 5.156 |

# CHAPTER 4

## FLOW PAST 1 AND 5 CYLINDERS; FULL DOMAIN SIMULATION

In the present chapter, the flow field for a single or an array of cylinders is determined using Darcy's law and is discussed in Chapter 2. There is no separation of flow and there is no importance for the hydrodynamic boundary layer. Here we concentrate only on the thermal boundary layer and associated heat transfer during the transient as well as steady state.

### 4.1 HEAT TRANSFER FROM A SINGLE CYLINDER

For a single cylinder, we compare the results obtained by finite element method, operator splitting, operator splitting with universal limiters and boundary layer method. For low Peclet number the domain size is 6 x 6 units, number of elements is 256 and number of nodes is 576. For high Peclet numbers the domain size is the same the number of elements is 576, the number of nodes is 1248 and the mesh is fine. Here we find the mesh size by calculating the minimum thermal boundary layer thickness and ensure that within this thickness, we have sufficient number of elements. Time steps used in present case for $Pe \leq 10$ is 0.095, for $Pe \geq 10$ is 0.05 and 0.03. For $Pe = 50$, and $Pe = 100$ we used two different types of grids (See Figures 4.1 and 4.2).

At low Peclet numbers all methods give identical answers at steady state except the boundary layer method (Figures 4.3 and 4.4). At higher Peclet number flows the classical finite element solution deviates from the other methods (see Figures 4.5 and 4.6). The boundary layer method is invalid at low Peclet number. Its predictions match numerical results at intermediate Peclet numbers. At high Peclet numbers the local Nusselt number predicted using boundary layer approximation fails to show the dip observed with numerical schemes. This issue is discussed again, later. The average Nusselt number at small time is large and decreases rapidly to reach steady state. For a constant heat flux

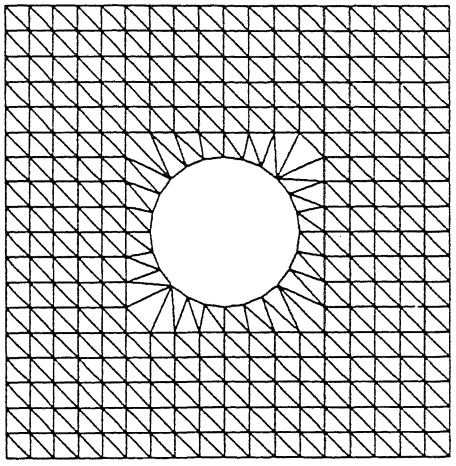


Figure 4.1 Finite Element Mesh with Single Cylinder (type 1)

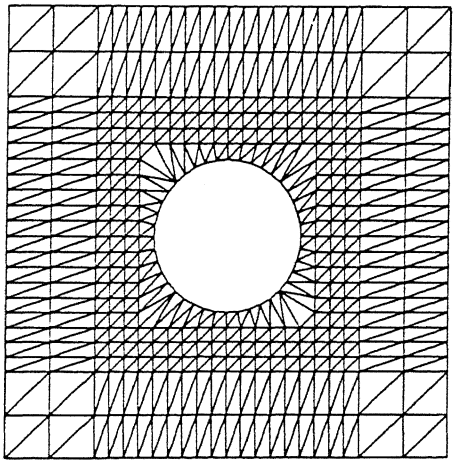


Figure 4.2 Finite Element Mesh with Single Cylinder (type 2)

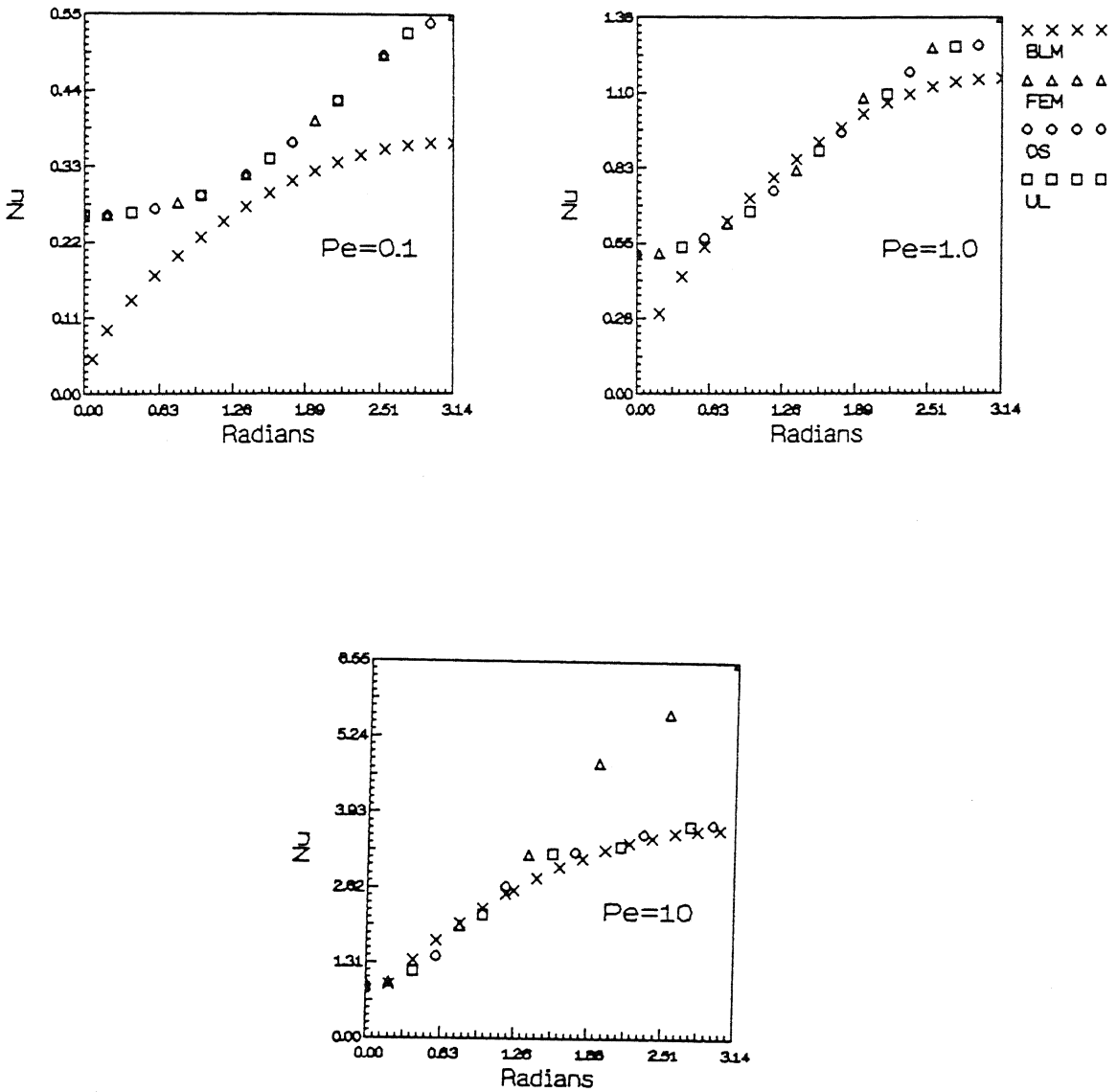



Figure 4.3 Variation of Steady Nusselt Number on the Cylinder Surface for Pe = 0.1, 1 and 10

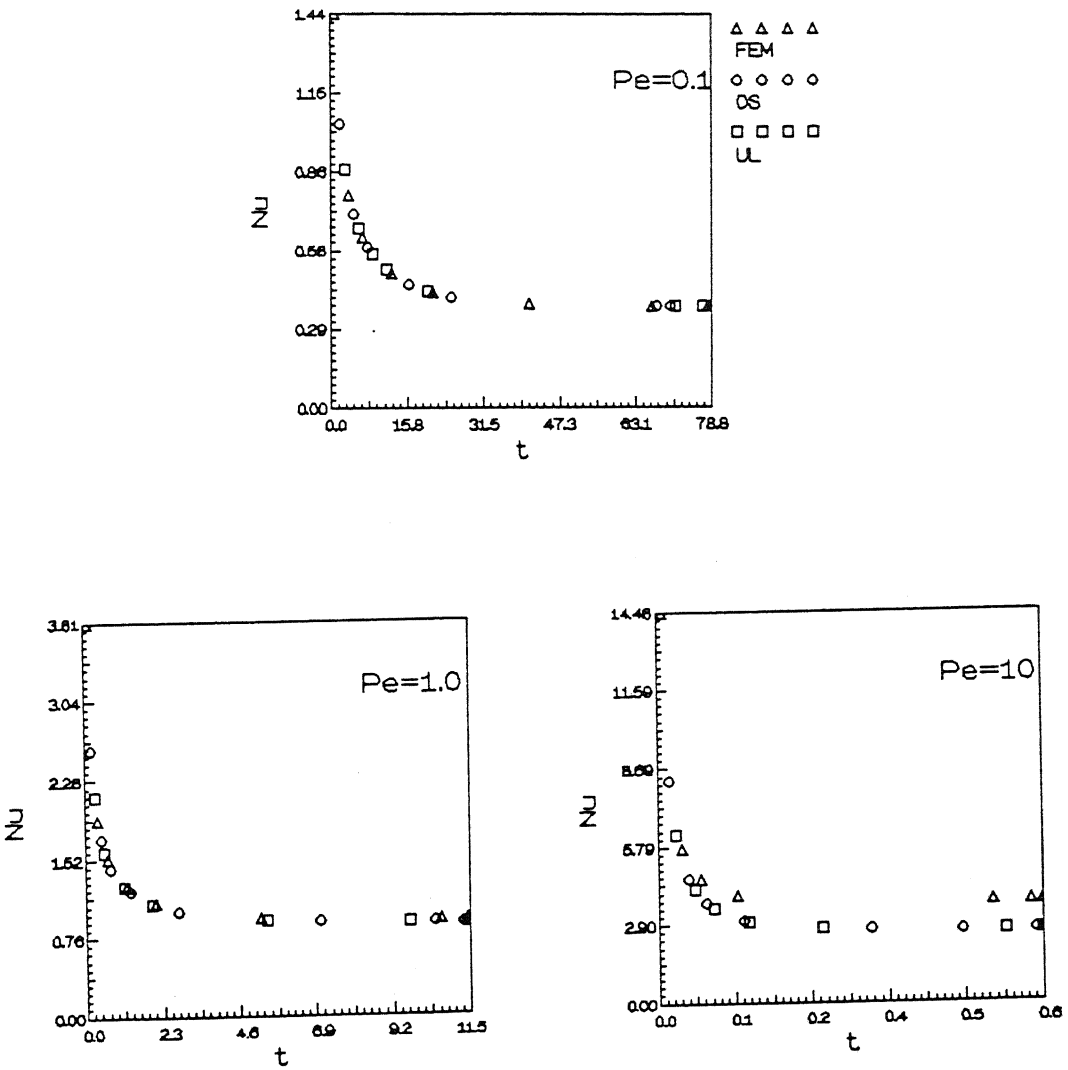



Figure 4.4 Variation of Average Nusselt Number with time for Pe = 0.1, 1 and 10

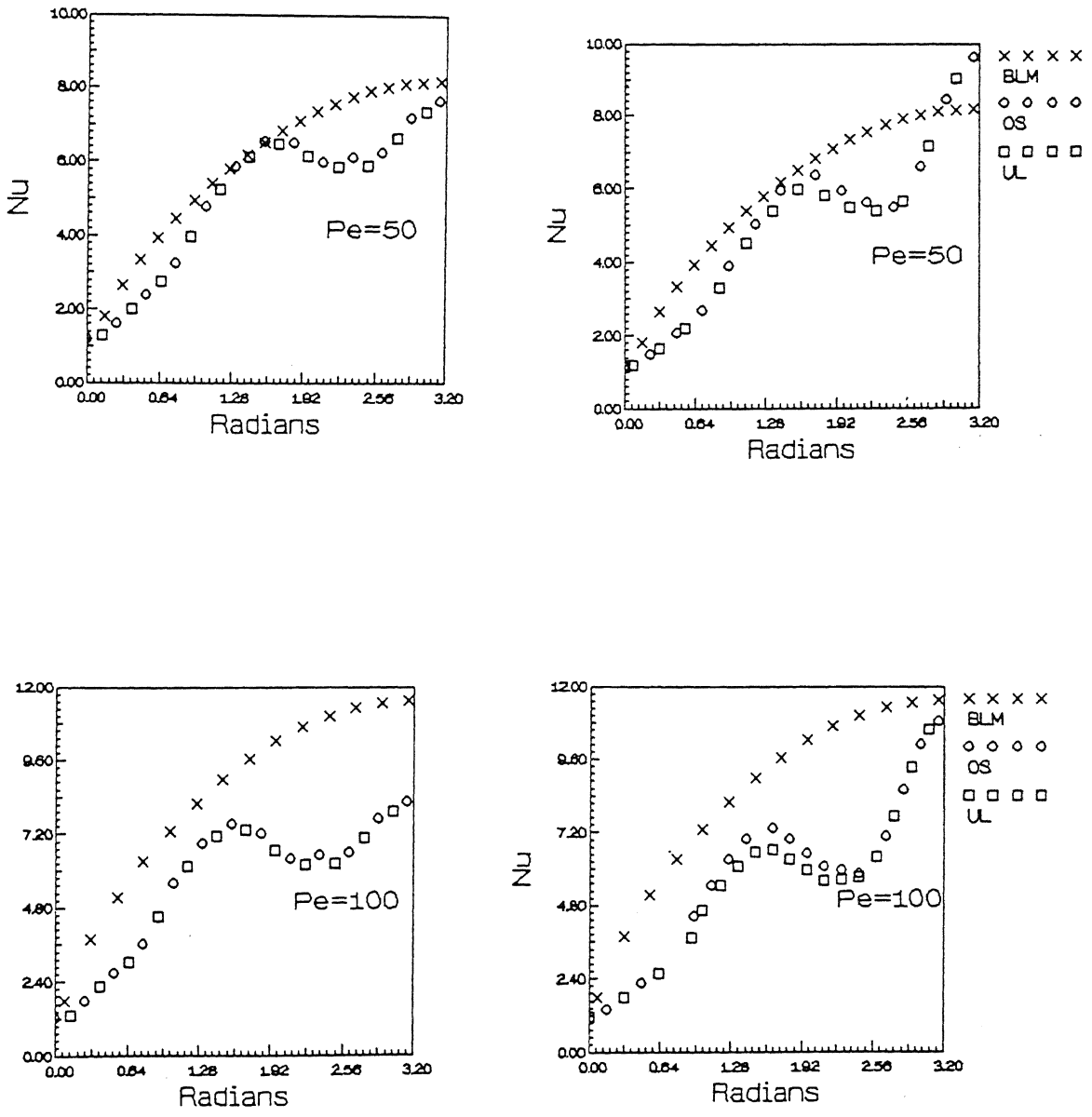



Figure 4.5 Variation of Steady Nusselt Number on the Cylinder Surface for Pe = 50 and 100

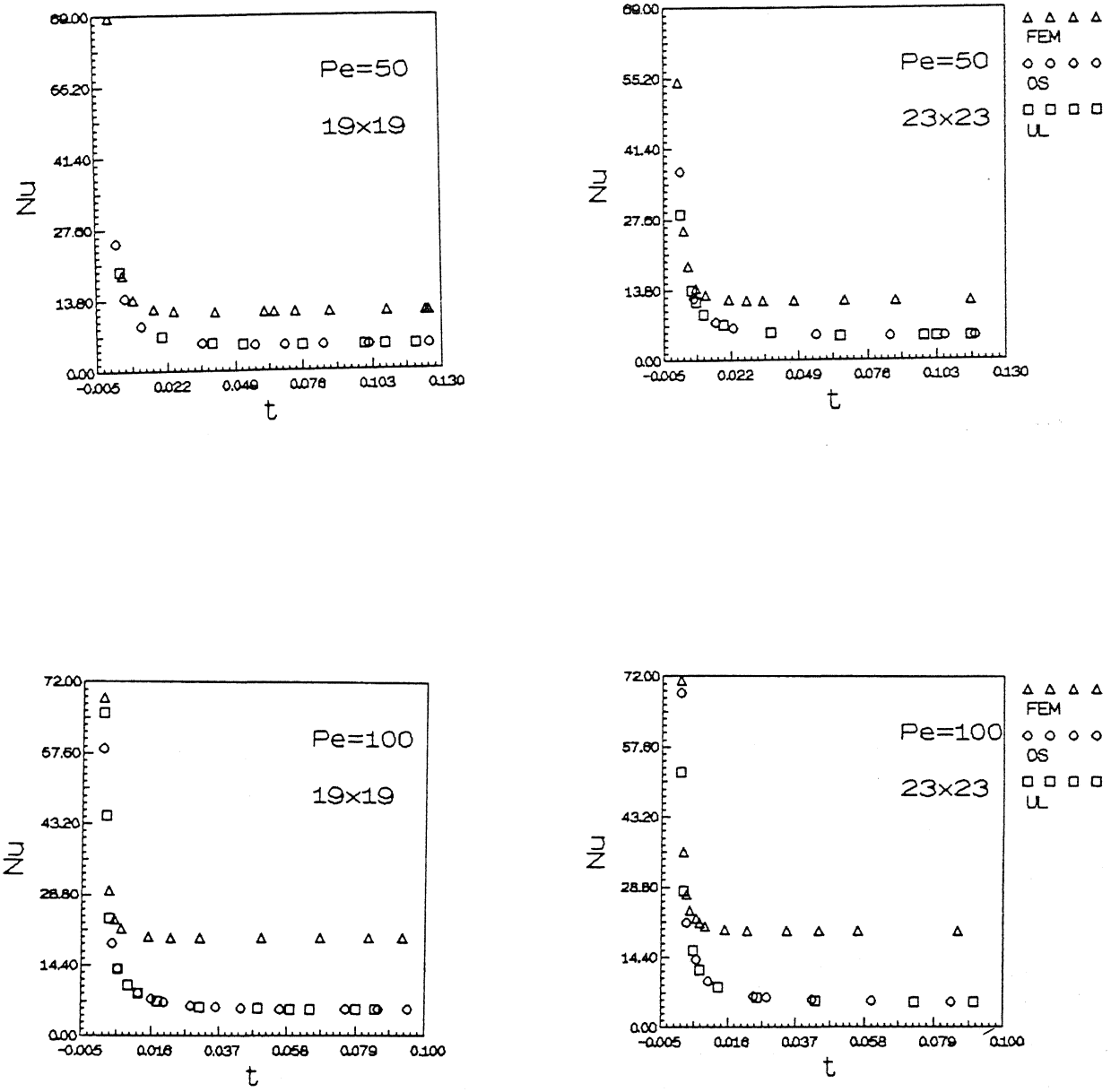



Figure 4.6 Variation of Average Nusselt Number with time for Pe = 50 and 100

specified over the cylinder surface, $Nu_{local} = 1/T$, where T is the local wall temperature. Initially the average temperature of the cylinder is small and increases gradually with time. Time shown in plots for average Nusselt number is defined as, $t = \alpha t/L^2$ and is the non-dimensionalized time scale.

At low Peclet numbers the steady average Nusselt number is low compared to its value at high Peclet number. This is because at high Peclet number convection dominates transport. A greater amount of heat is lost from the cylinder and the temperature of cylinder decreases. At low Peclet number ($Pe \leq 10$) the finite element method, operator splitting and operator splitting with universal limiters coincide (Figure 4.4). At high Peclet numbers (10, 50, 100), the finite element method deviates from operator splitting and operator splitting with universal limiters (See Figure 4.6). For completeness, the no flow problem, Pe = 0.0 is also presented here (Figure 4.7). There is no steady state in the absence of flow and Nusselt number goes to zero at long time.

## 4.2 HEAT TRANSFER FROM FIVE CYLINDERS

In case of 5 cylinders, a boundary layer solution is not possible to account for the inter-cylinder influence. Instead, we have compared the three methods namely finite element method, operator splitting and operator splitting with universal limiters. The domain size is 11 x 11 units. The number of elements is 808 and the number of nodes is 1780. The five cylinders are located at (5.5, 5.5), (2.5, 2.5), (2.5,8.5), (8.5,2.5), (8.5,8.5), as seen in Figure 4.8. The maximum allowable time step is calculated using the Courant number condition. The time step is calculated as $\Delta t = 0.09$ for all the three Peclet numbers namely, 0.1, 1 and 10.

The thrust of the calculation is to determine the effect of the 4 surrounding cylinders on the central cylinder. At Peclet numbers of 0.1 and 1 the steady Nusselt number value is less than that a single cylinder but at Pe = 10 the difference is small (See Figure 4.9). The reason for this behaviour is the following at low Peclet number, diffusion is main mechanism of transport of energy. All 5-cylinders have a constant heat flux boundary

condition. Due to the other cylinders located around the central cylinder, the temperature gradient decreases and the heat transfer from central cylinder reduces. This in turn affects the steady Nusselt number of this arrangement.

From Figure (4.10), variation of steady Nusselt number with angle is comparable upto Pe = 10 with all three methods. Simulation with higher Pe was not possible owing to memory limitation on the HP system.

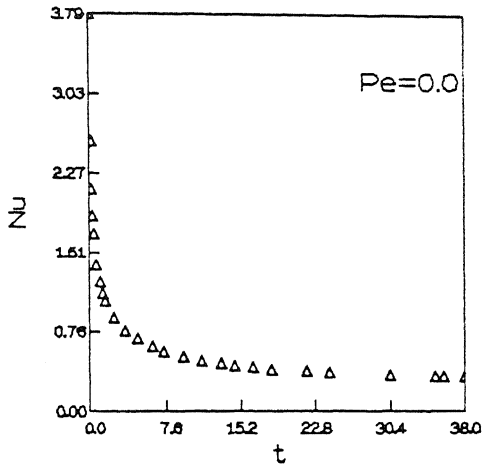



Figure 4.7 Variation of Average Nusselt Number with time for Pe = 0

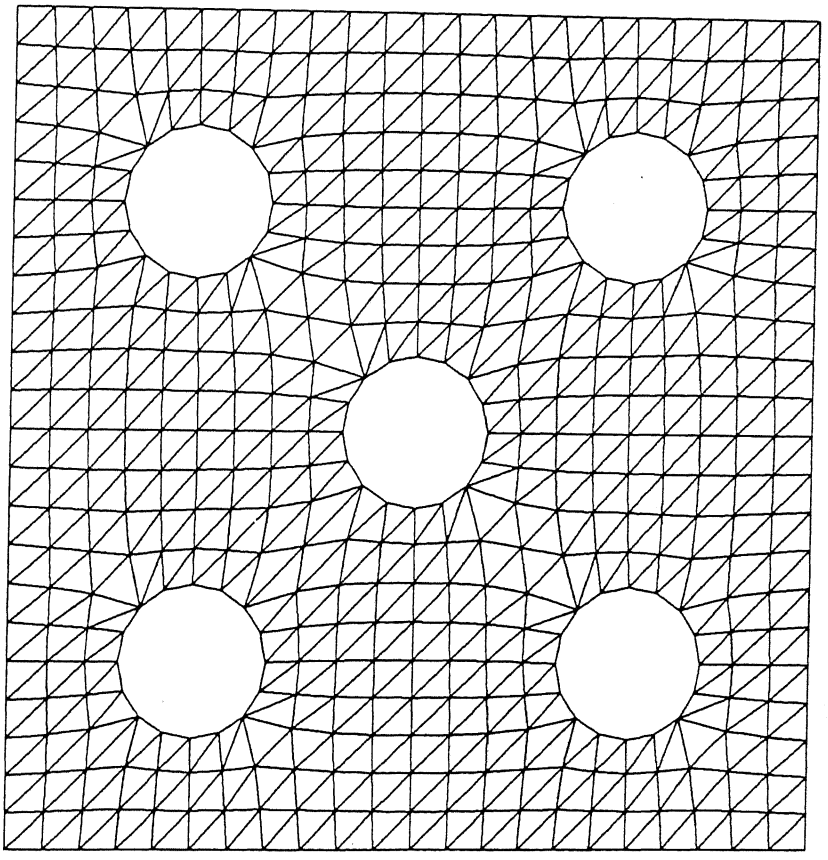


Figure 4.8 Finite Element Mesh of the Computational Domain with Five Cylinders

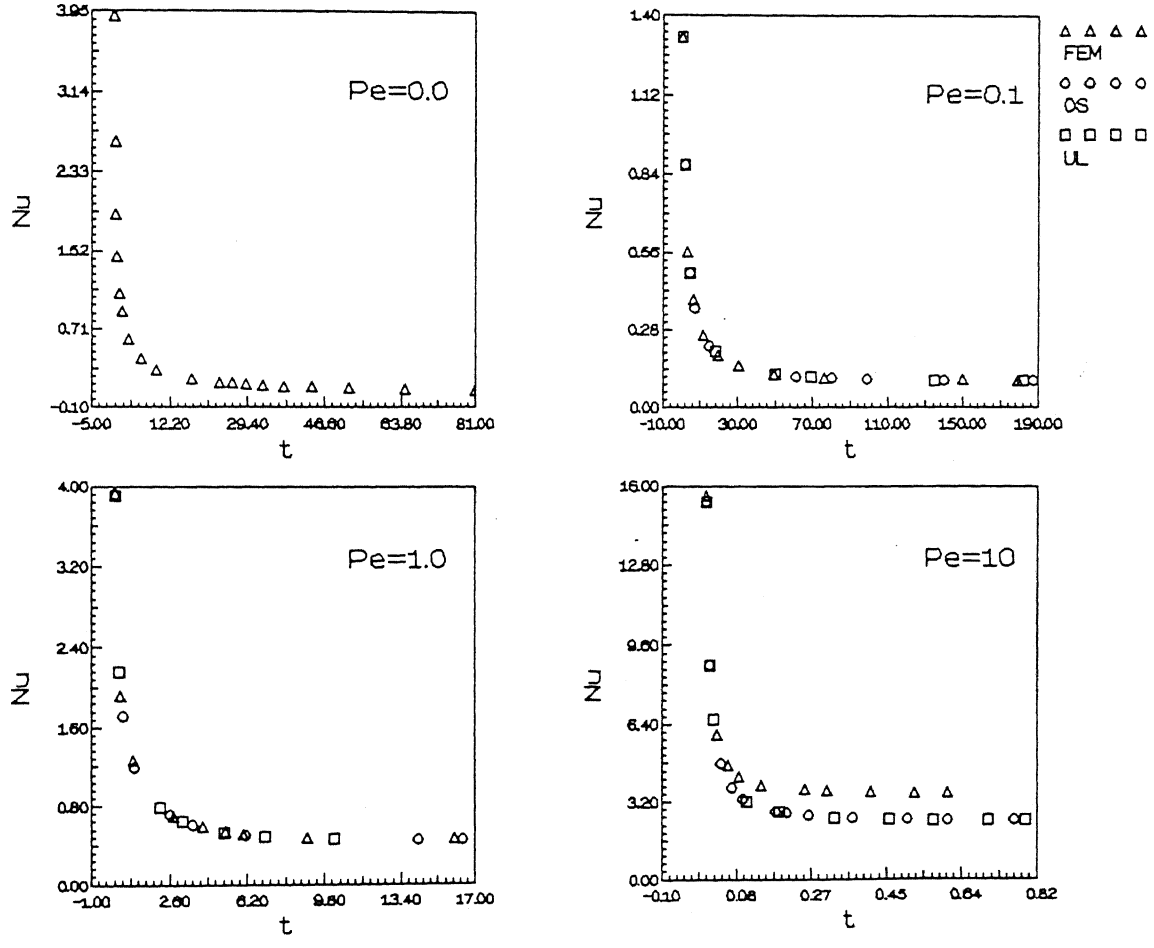



Figure 4.9 Variation of Average Nusselt Number with time for Pe = 0.1, 1 and 10 for the Domain with Five Cylinders

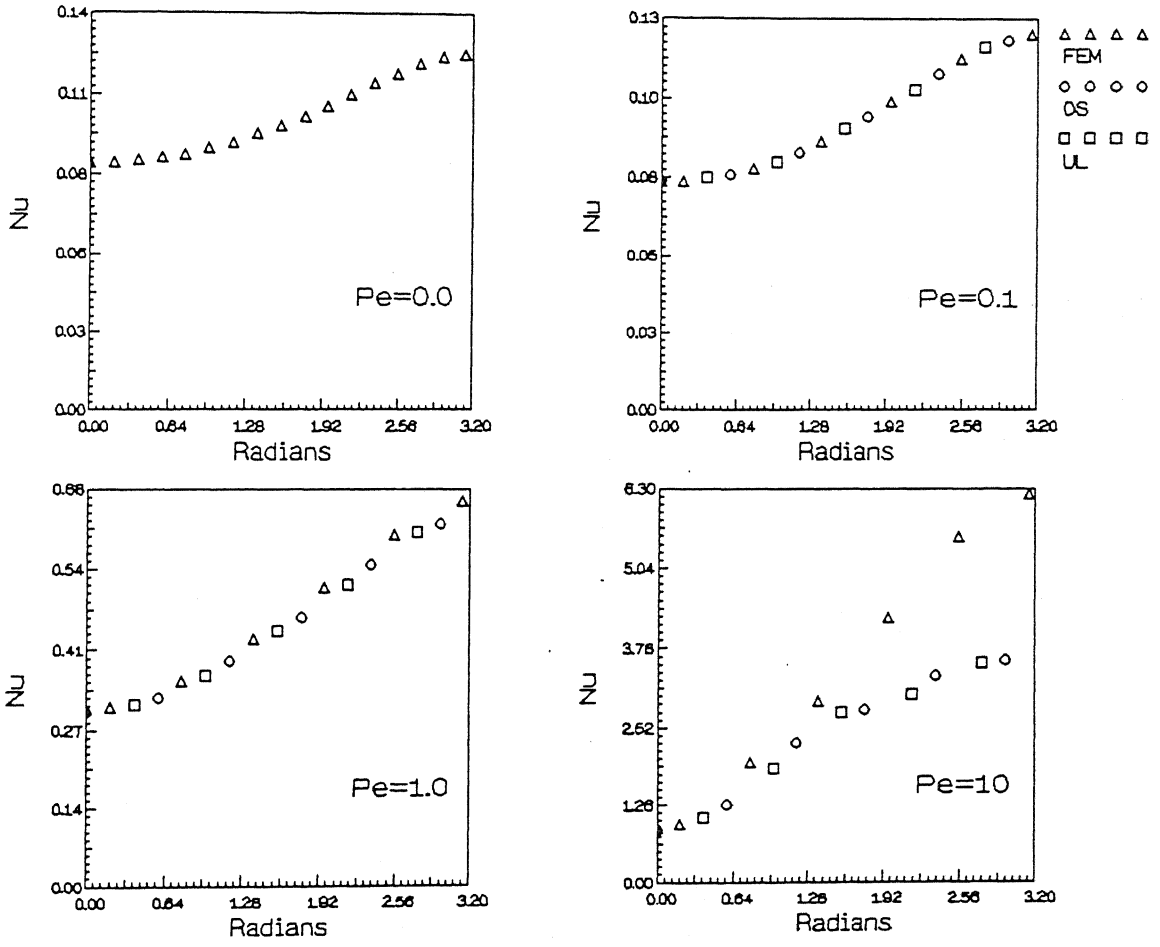



Figure 4.10 Variation of Steady Nusselt Number on the Cylinder Surface for Pe = 0.1, 1 and 10 for the Domain with Five Cylinders

# CHAPTER 5

# RESULTS OBTAINED USING DOMAIN DECOMPOSITION

## 5.1 MOVEMENT OF FRONT PROBLEM

In the present study, the domain divided into 2, 4 and 9 subdomains has been considered. The formulation of domain decomposition is discussed in Chapter 2. The full domain has the size of 18 units length, 6 units width. The time step used is 0.5 for all Peclet numbers. With 2-subdomains we treat only one interface and so it is faster than the computations for 4 and 9 subdomains. In this present case results have been compared with the analytical solution. Agreement between analytical and domain decomposition is excellent (See Figures 5.1, 5.2, 5.3, 5.4, 5.5 and 5.6). The efficiency of domain decomposition calculation also depends upon the type of division. In case of 2-subdomains, the partition is meaningful since the transport of energy from left to right. In case of 4 and 9, subdomains the domain is divided horizontally as well without a meaningful physical mechanism being present in this direction. Convergence rates are fast only in case of physically meaningful division of the domain. These are clear from Table 5.1, where CPU time is given. The CPU times for domain decomposition is more compared to full domain calculations. If these calculations were to be implemented on a parallel computer, the CPU time is expected to fall below that for a full domain.

## 5.2 FLOW PAST A SINGLE CYLINDER, EFFECT OF OUTFLOW PLANE

The outflow boundary condition used in the present work is $\frac{\partial T}{\partial n} = 0$, where n is the outward drawn normal at that plane. This condition means that, temperature from that plane onwards is constant along the streamline direction. This boundary condition is valid if the outflow plane is faraway from the cylinder surface. Solving the governing equation over large domains on existing computers is a problem. Domain decomposition is a

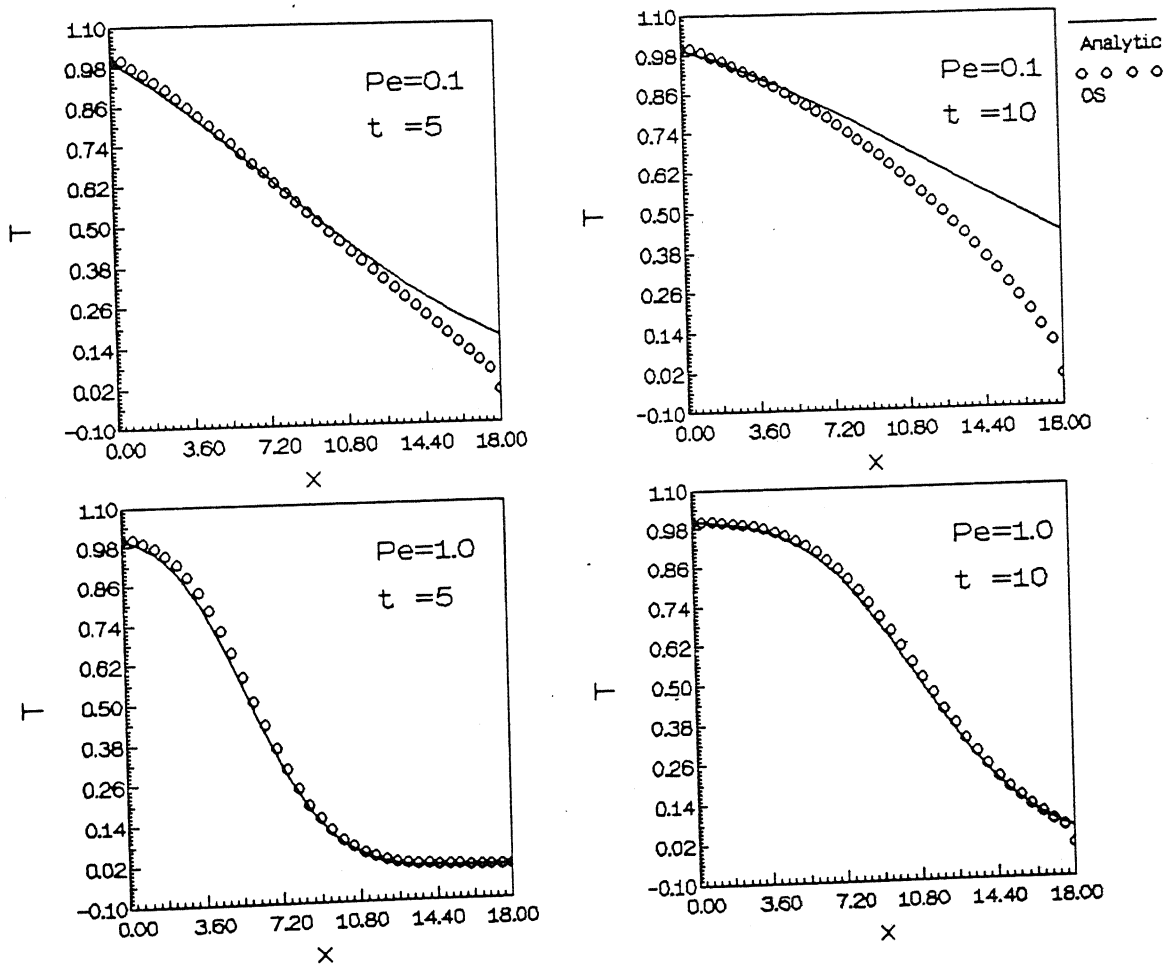



Figure 5.1 Variation of Temperature along Flow Direction with Two Subdomains for Pe = 0.1 and 1

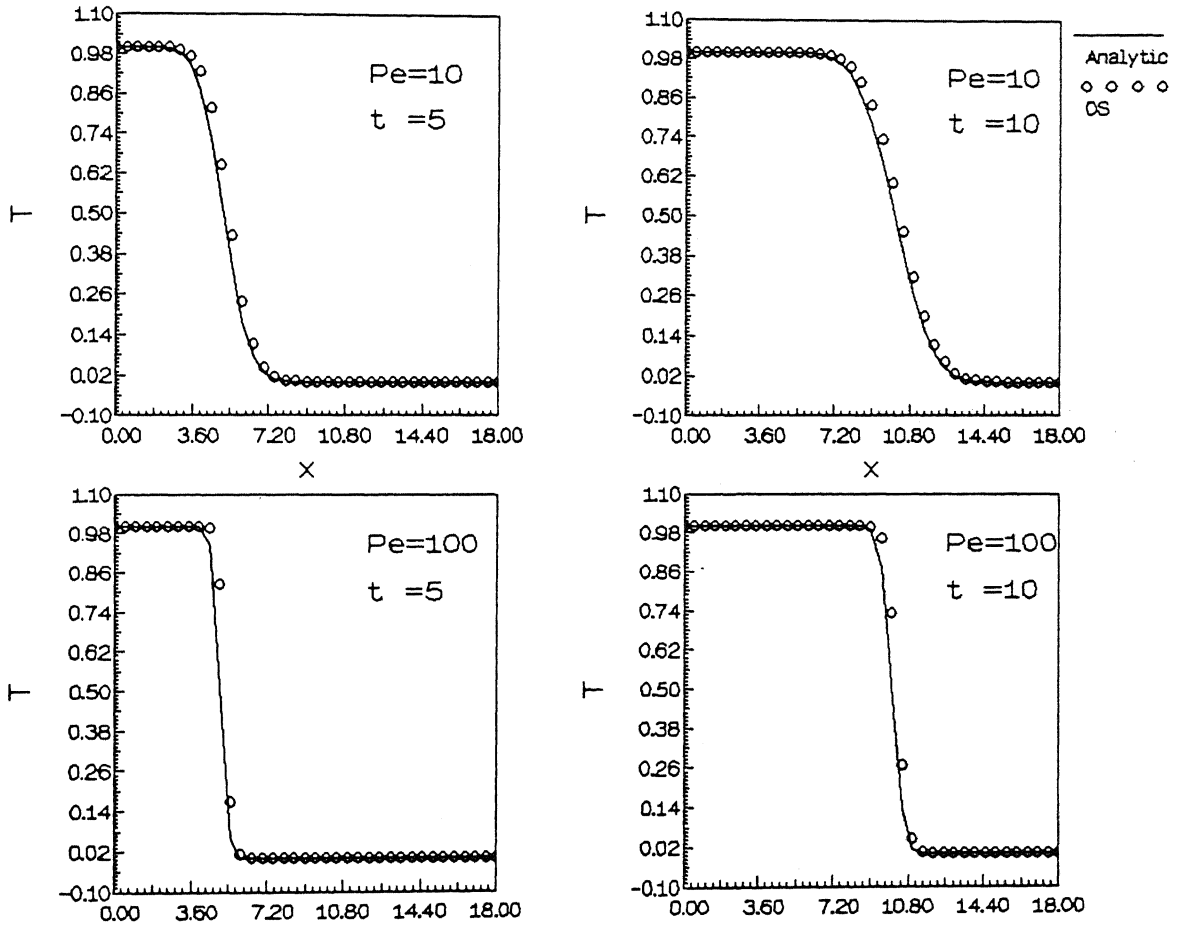



Figure 5.2 Variation of Temperature along Flow Direction with Two Subdomains for Pe = 10 and 100

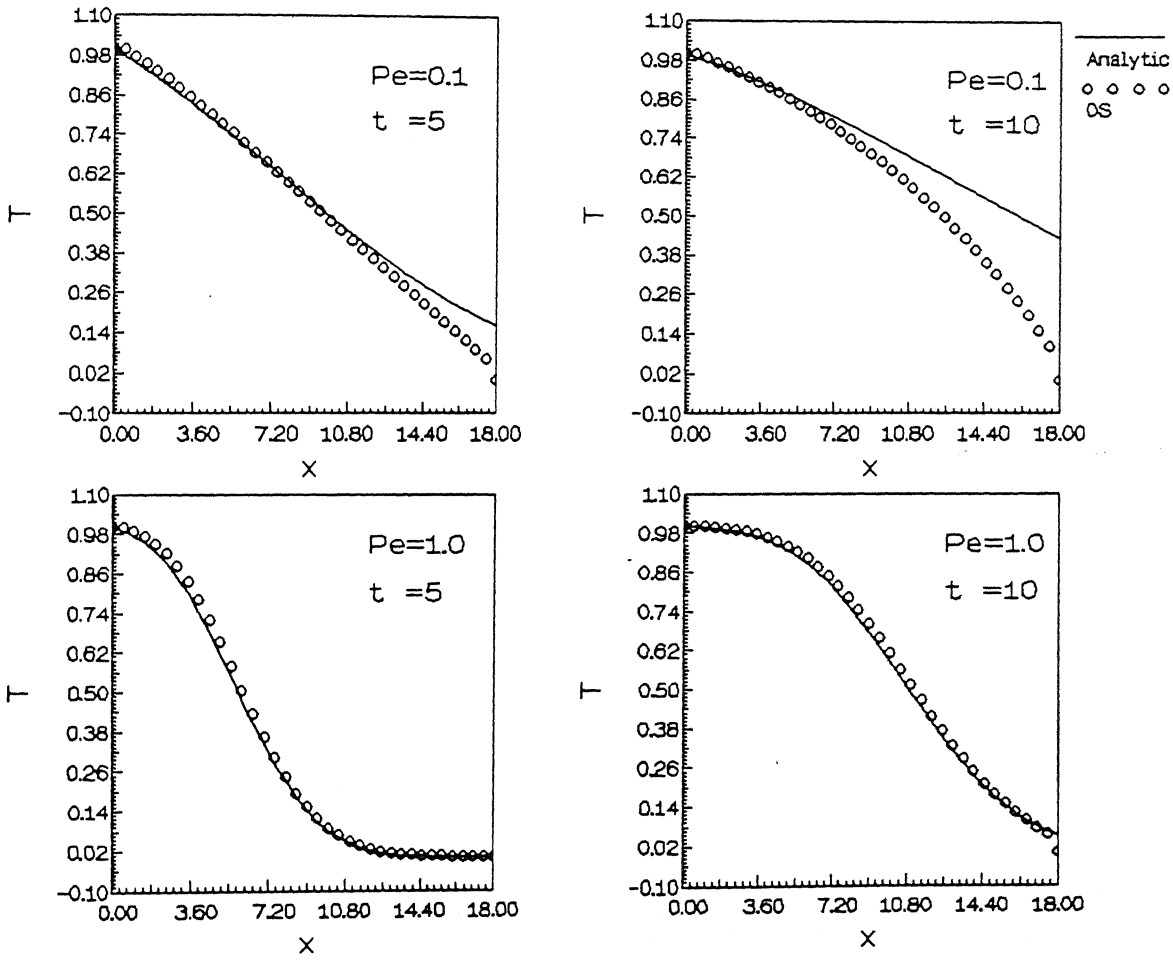



Figure 5.3 Variation of Temperature along Flow Direction with Four Subdomains for Pe = 0.1 and 1

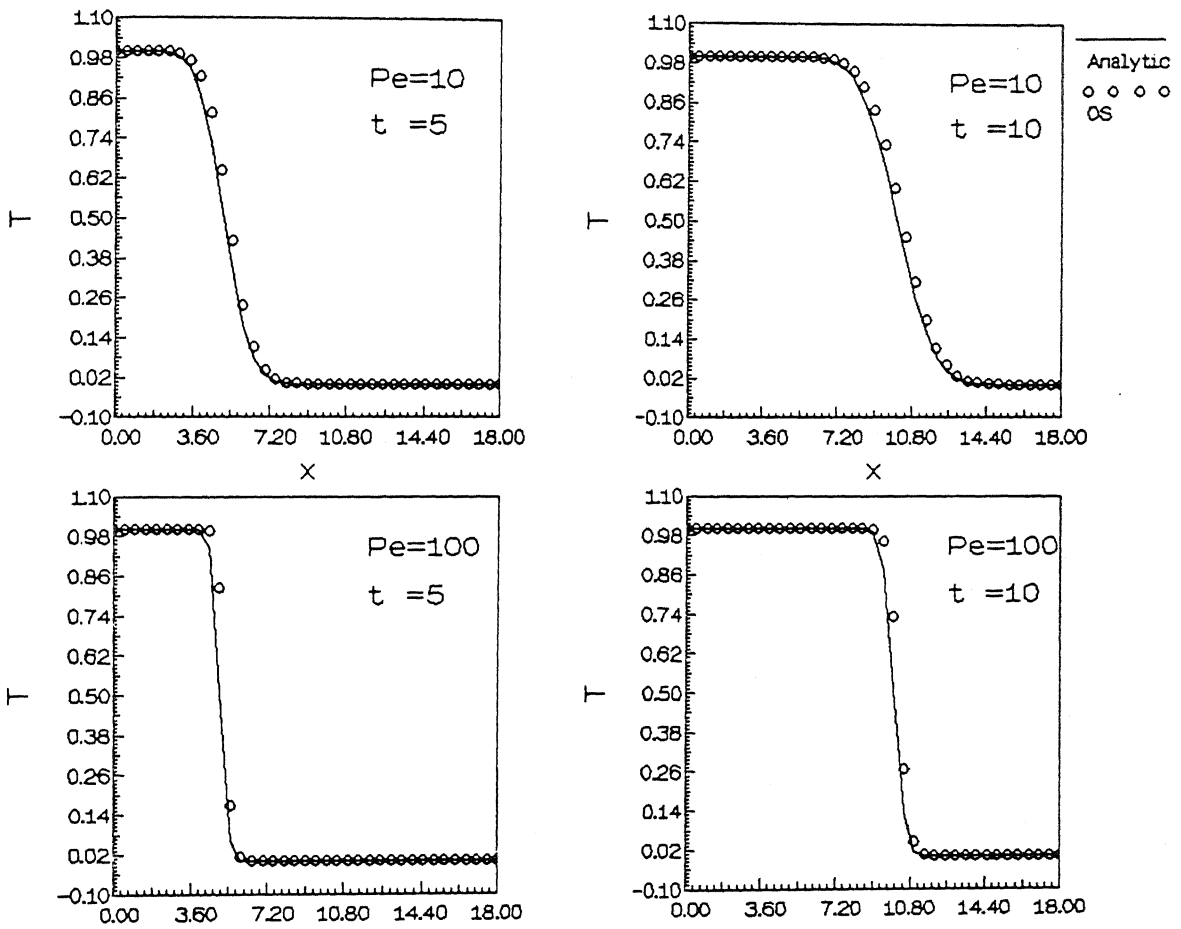



Figure 5.4 Variation of Temperature along Flow Direction with Four Subdomains for Pe = 10 and 100

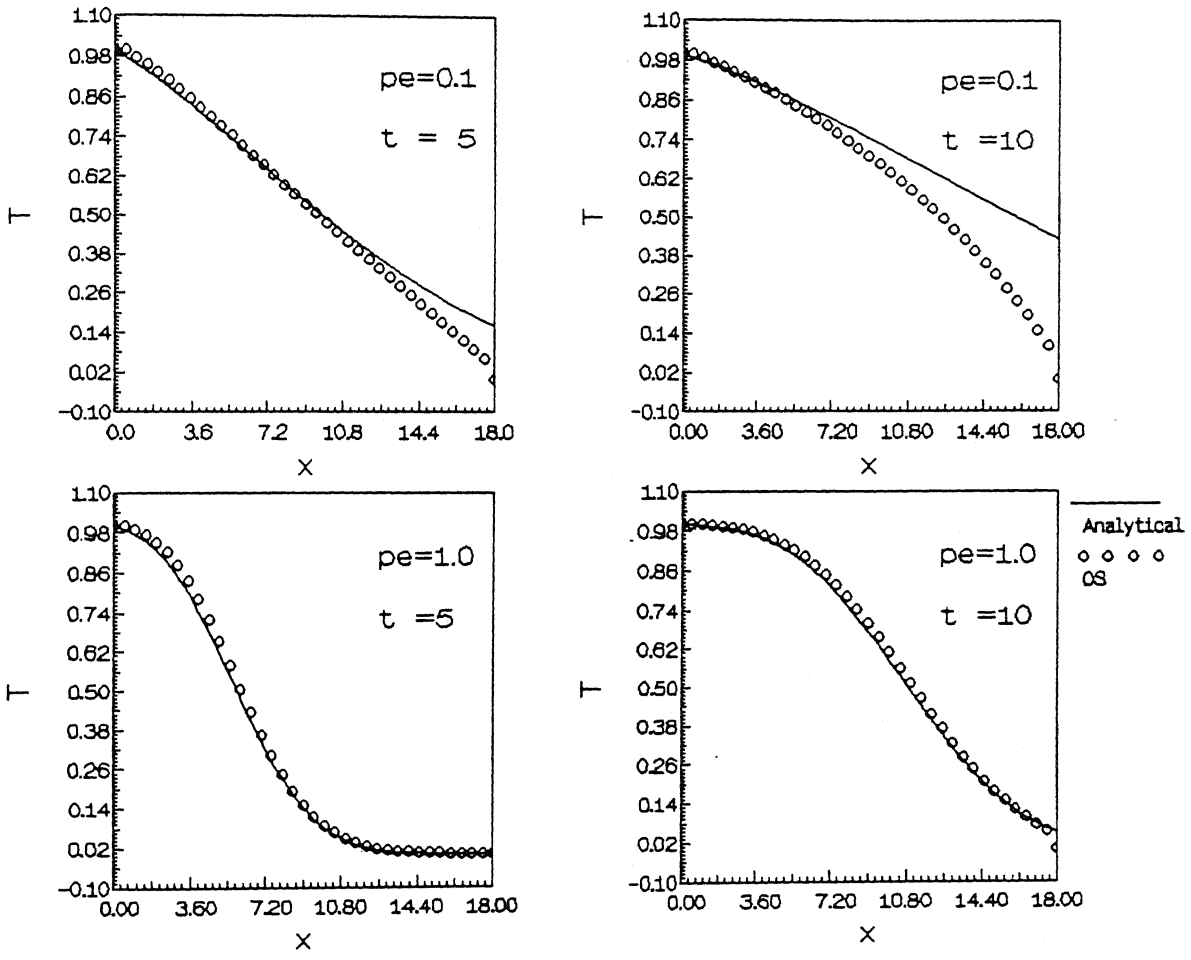



Figure 5.5 Variation of Temperature along Flow Direction with Nine Subdomains for Pe = 0.1 and 1

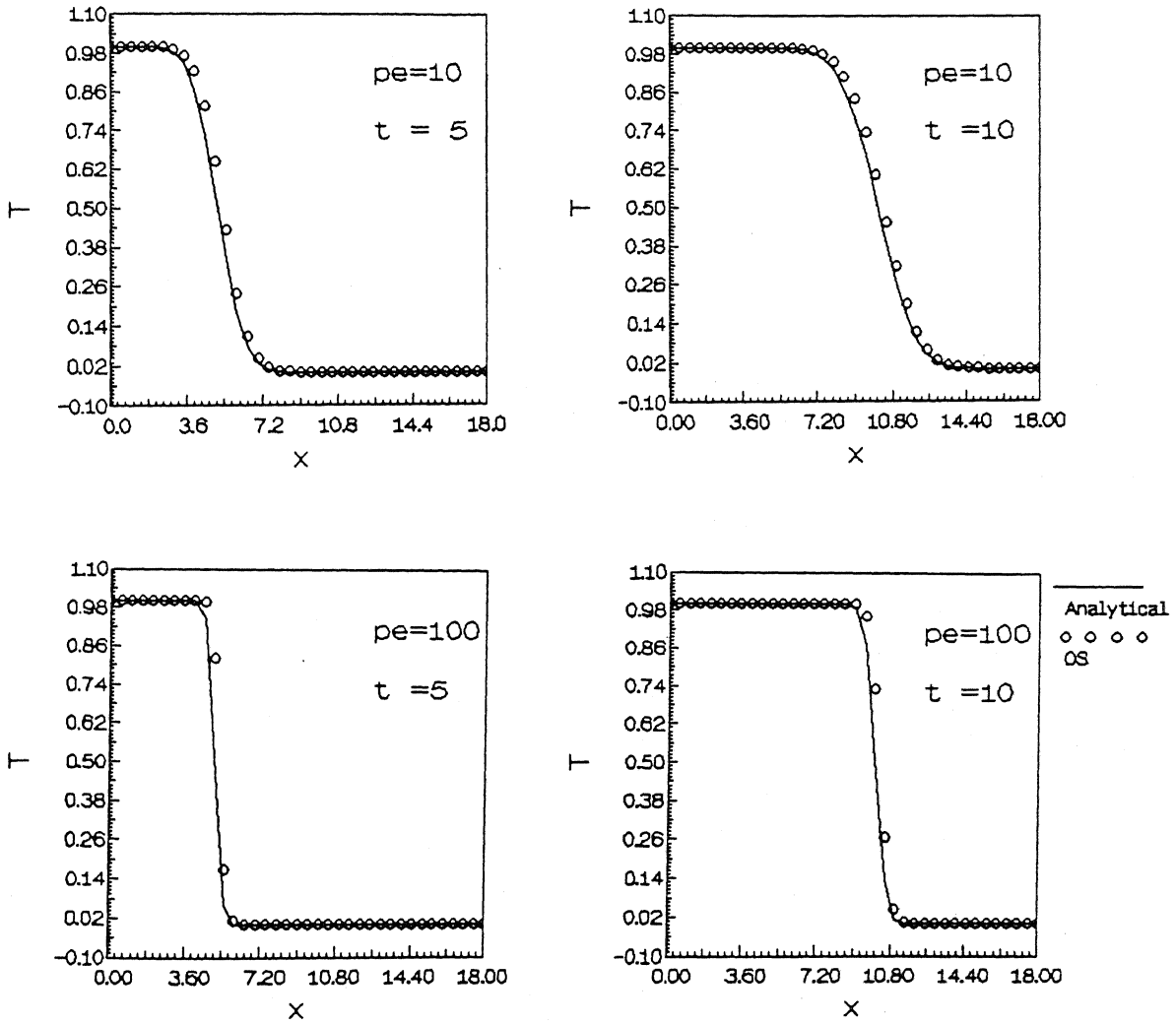



Figure 5.6 Variation of Temperature along Flow Direction with Nine Subdomains for Pe = 10 and 100

convenient approach available at this point to circumvent this difficulty.

Originally a domain of size 6 x 6 was used for a cylinder diameter of 2-units. Compared to this diameter a length of 6 units is not large enough. This results in spurious reflection of errors from the outflow plane. At low Peclet numbers the transport of energy is due to diffusion and there is no effect of the outflow plane. In case of high Peclet number flows the effect of outflow plane is greater, because transport of energy is predominantly due to convection.

From Table 5.2 the effect of the location of the outflow plane is clear enough.

For the two domain problem, the domain size is 12 x 6. For low Peclet numbers, 544 elements and 1276 nodes and for high Peclet number 1224 elements and 2580 nodes have been used. Figure (5.7) shows Nusselt number as a function of angle $\theta$ for a domain of size 6 x 6 and another for a domain of size 12 x 6 units with 2-subdomains each having 6 x 6 units of region size. For Pe < 10 both curves coincide showing that the outflow location is not critical. For Pe > 10, the curves deviate from each other but the overall variation is the same. From this figure, the outflow effect is clear, because there is a change in steady Nusselt numbers over the cylinder surface. Figure 5.8 shows the variations of Nusselt number with time. The isotherms patterns are shown in Figure 5.9. From this figure, it can be observed that at low Peclet numbers the outflow condition is valid and at high Peclet numbers the outflow condition is not exactly satisfied.

## 5.3 9 - SUBDOMAIN SIMULATION FOR FLOW PAST A CYLINDER

In this section, we present results for a domain divided into 9 sub-domains with one cylinder at the centre of the domain (See Figure 5.10). The formulation of domain decomposition is discussed in Chapter 2. The full domain has the size of 6 x 6 units. The number of elements is 256 and number of nodes is 576 for Pe = 0.1, 1 and 10. The time step used is 0.1. Here we compare the variation of steady Nusselt number with angle $\theta$ with that of the full domain. At low Peclet numbers the variation of steady

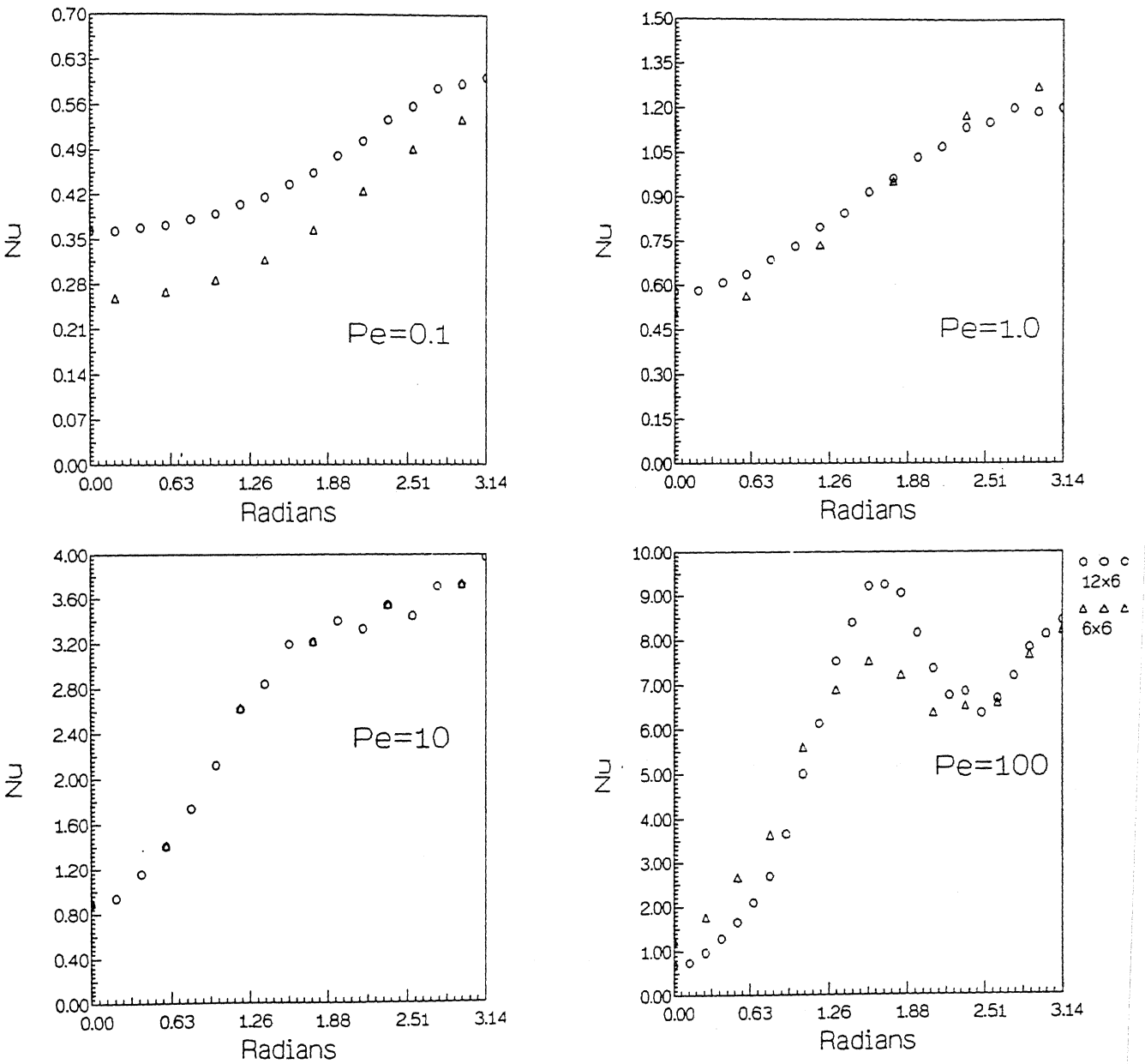



Figure 5.7 Variation of Steady Nusselt Number on the Cylinder Surface : Outflow Effect

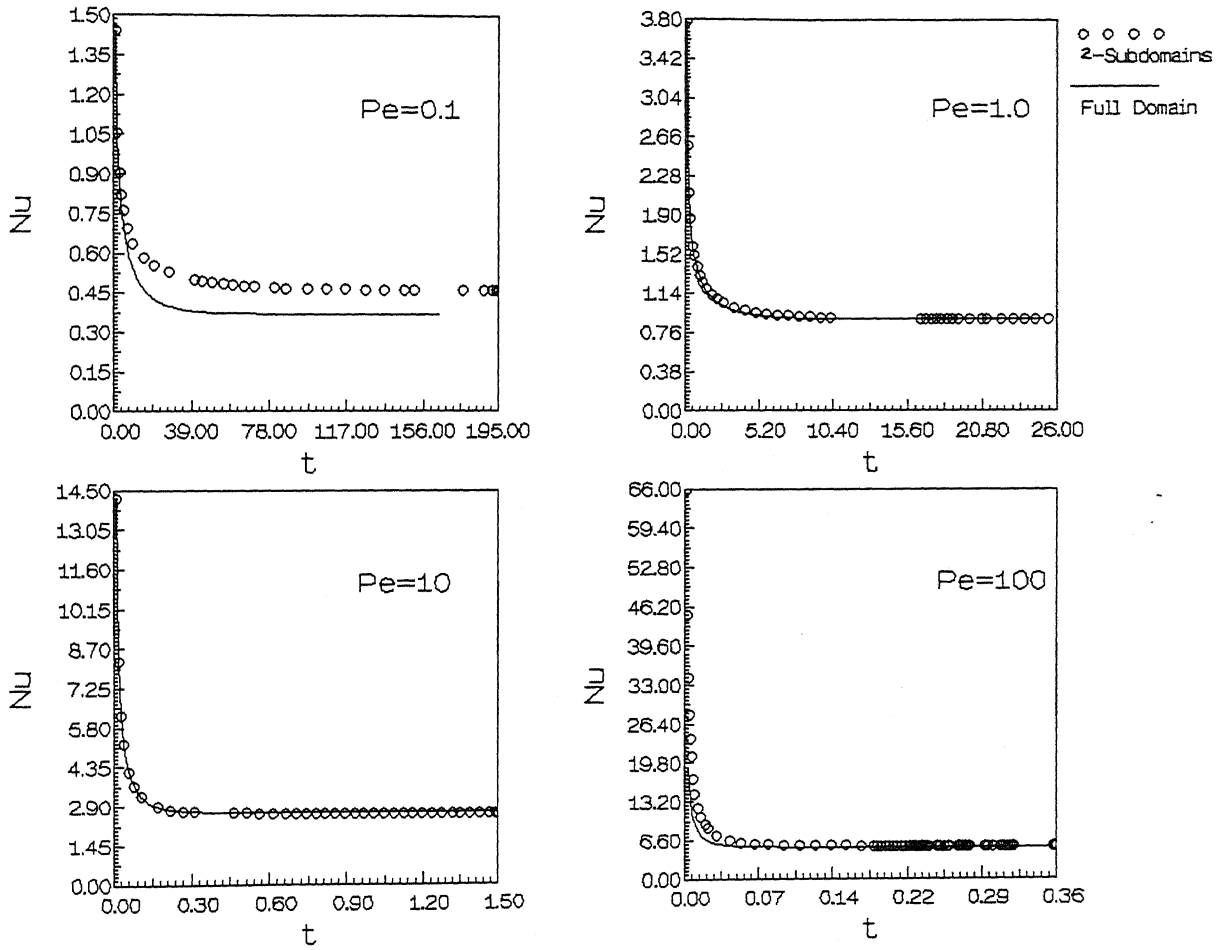



Figure 5.8 Variation of Average Nusselt Number with time : Outflow Effect

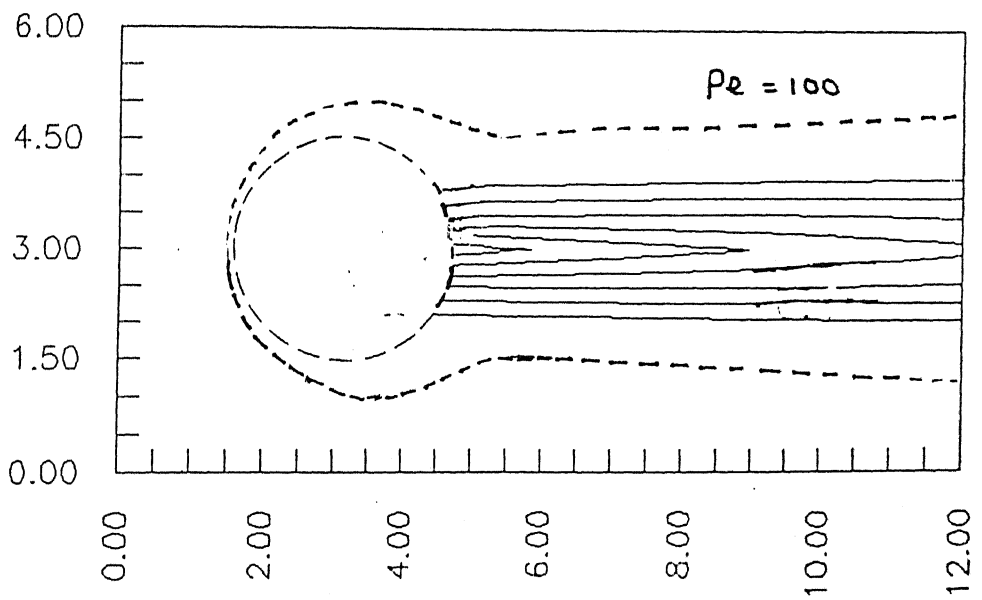



Figure 5.9 Isotherm Pattern(s) for Pe = 100 : Outflow effect

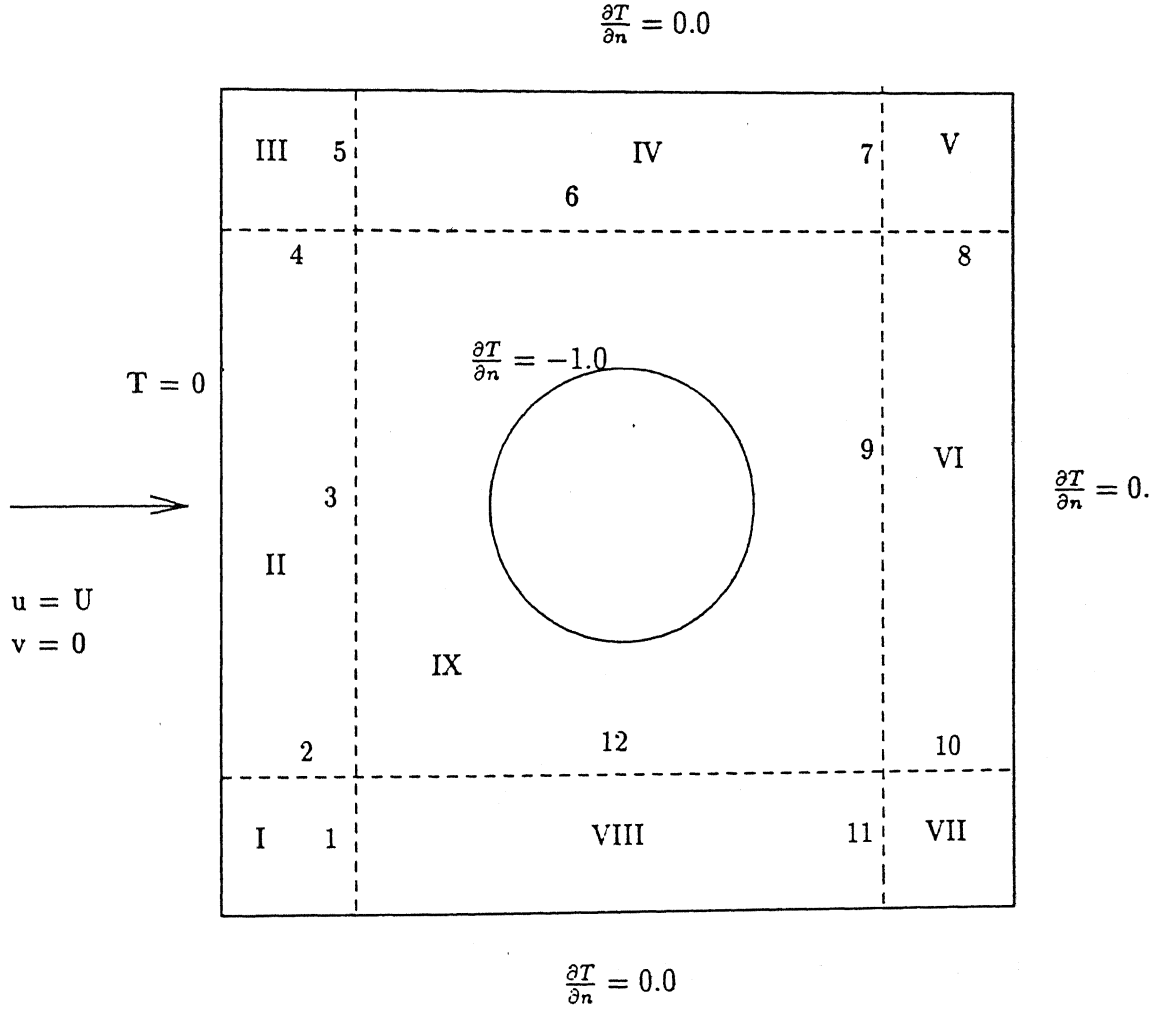



Figure 5.10 Schematic Diagram of Computational Domain divided into Nine Subdomains

Table 5.1 CPU Times in Minutes for Interface Error = 1.0e-04 : Without Cylinders

| Peclet Numbers | Full domain | 2-Subdomain | 4-Subdomain | 9-Subdomain |
|---|---|---|---|---|
| 0.1 | 1.828 | 1.18 | 1.26 | 1.03 |
| 1.0 | 1.74 | 1.85 | 3.20 | 2.98 |
| 10 | 2.35 | 3.33 | 9.66 | 9.00 |
| 100 | 2.28 | 10.91 | 42.21 | 43.66 |

Table 5.2 CPU Times in Minutes for Interface Error = 1.0e-06 : Without Cylinders

| Peclet Numbers | Full domain | 2-Subdomain | 4-Subdomain | 9-Subdomain |
|---|---|---|---|---|
| 0.1 | 1.828 | 1.51 | 4.25 | - |
| 1.0 | 1.74 | 3.85 | 10.85 | 8.81 |
| 10 | 2.35 | 10.01 | 26.28 | 26.7 |
| 100 | 2.28 | 32.70 | 113.91 | 115.9 |

Table 5.3 Steady Nusselt Number Values for Different Finite Element Meshes

| Method | Pe = 0.1 | Pe = 1.0 | Pe = 10 | Pe = 100 |
|---|---|---|---|---|
| 6 x 6 | 0.370 | 0.893 | 2.668 | 5.382 |
| 12 x 12 | 0.368 | 0.896 | 2.662 | 5.544 |

Nusselt numbers with angle $\theta$ (See Figure 5.11) is identical to that of full domain. The variation of average Nusselt number with time level is compared in Figure 5.12. The comparison between full domain and 9 subdomain calculations is good. This is a good check on the domain decomposition. Hence it can be implemented for 5-cylinders, for large domains and for high Peclet number flows.

We present an explanation for drop in Nusselt Number around $\theta = 120^\circ$ for Peclet numbers greater than 50. The local Nusselt number on the surface of the cylinder is subjected to the conflicting influences of increase in boundary layer thickness in the flow direction and increase in velocity between $\theta = 90^\circ$ to $180^\circ$. The increase in the boundary layer thickness reduces the local Nusselt number in contrast to this, the increase in local velocity increases the local Nusselt number. The breakthrough between these two factors will occur at an angle that depends on the Peclet number. For Pe $\leq$ 10 the effect of boundary layer growth is dominant. For Pe $\geq$ 50 both effects are simultaneously present under non-monotonic variation in the local Nusselt number is seen. On the rear of the cylinder ($0^\circ \leq |\theta| \leq 90^\circ$). The flow is decelerated and Nusselt number drops at fairly rapid rate.

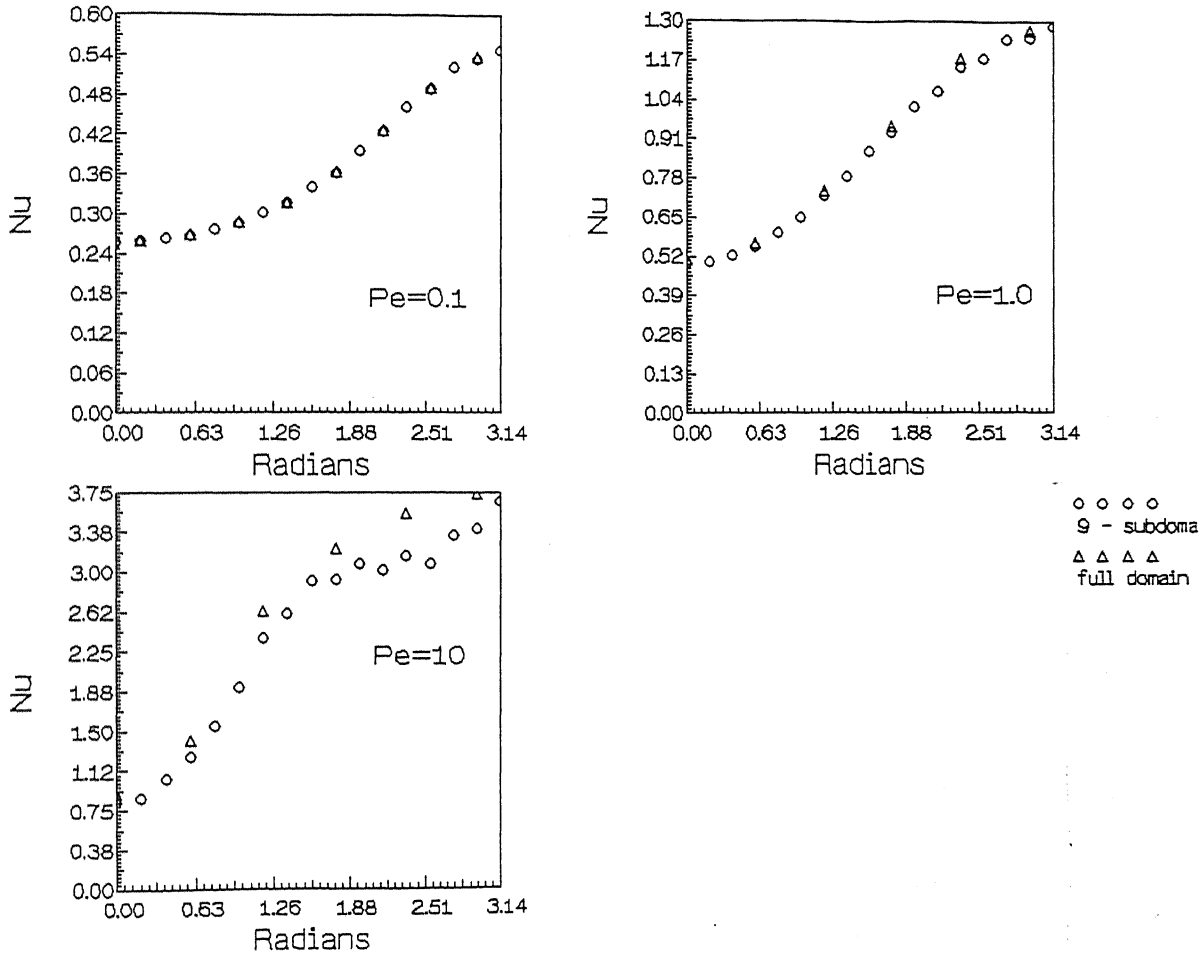



Figure 5.11 Variation of Steady Nusselt Number on Cylinder Surface for Pe = 0.1, 1 and 10 : Nine Subdomains

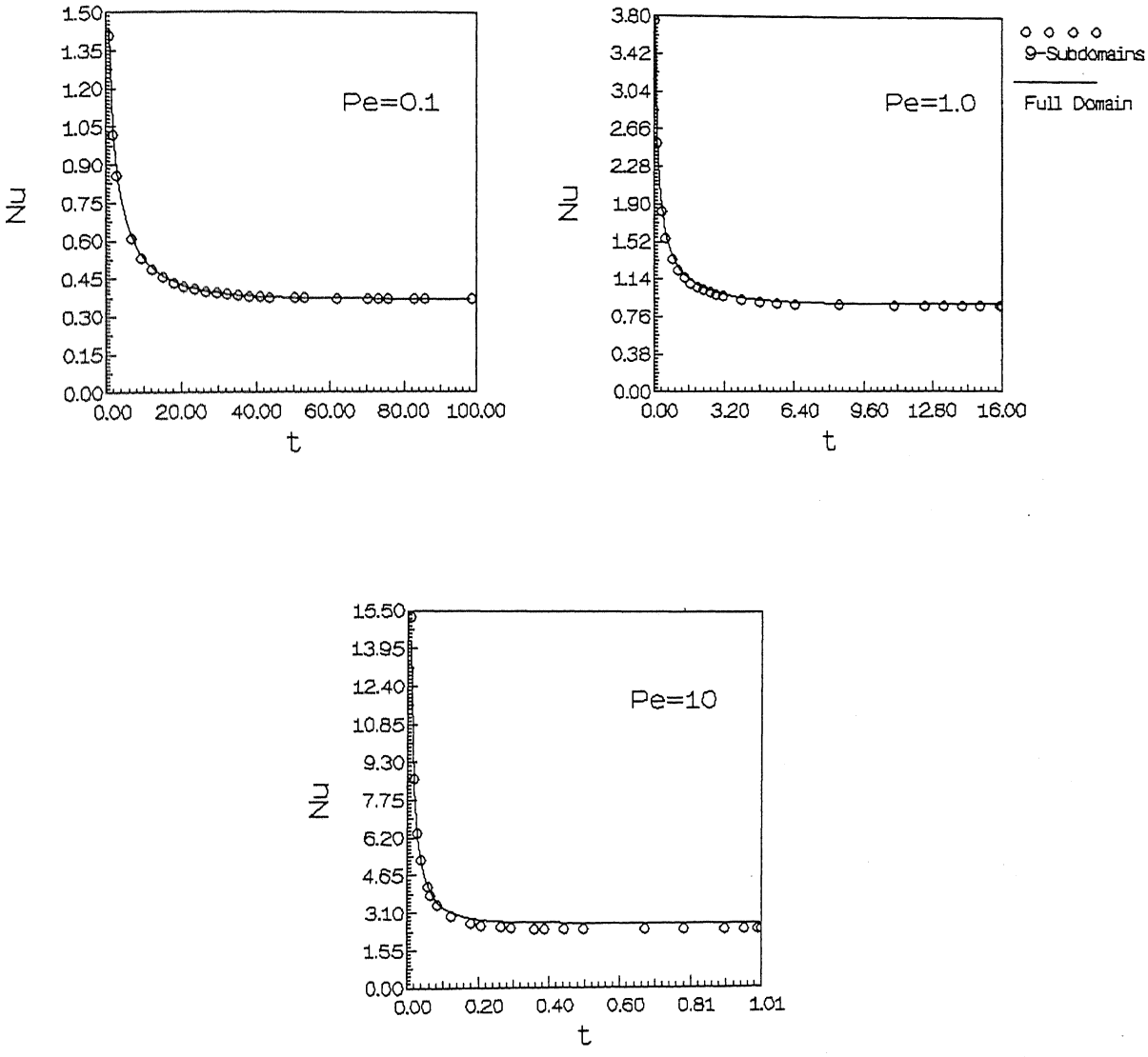
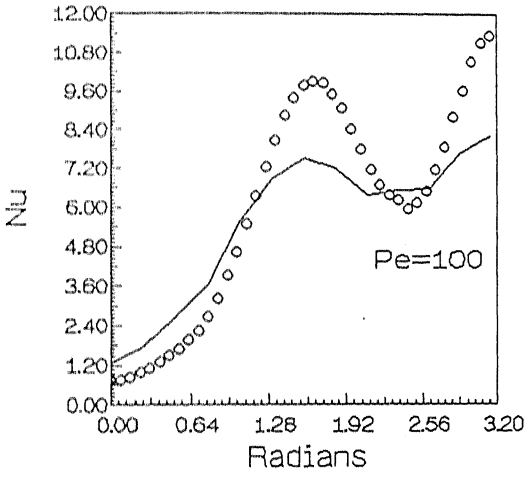



Figure 5.12 Variation of Average Nusselt Number with time for Pe = 0.1, 1 and 10 : Nine Subdomains



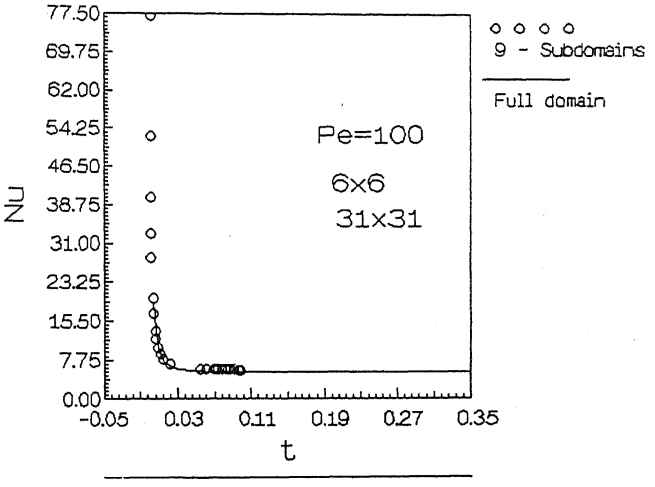




Figure 5.a Variation of Steady Nusselt Number on the Cylinder Surface for Pe = 100

Figure 5.b Variation of Average Nusselt Number with time for Pe = 100

# CHAPTER 6

## COMMENTS ON PARALLELIZATION

To take advantage of the inherent flexibility of the finite element method in solving for flows within complex geometry, it is necessary to produce an efficient implementation of the method. Functional splitting and domain decomposition are two methods which will make finite element method very efficient. Both functional splitting and domain decomposition are useful for parallelization of any problem.

Two types of parallel computers, those are based on architecture are widely used. Those are single instruction, multiple-data stream (SIMD) and multiple instruction, multiple-data stream (MIMD). The single instruction multiple-data stream computers are extensively used in scientific applications. Developing a computer code for single-instruction, multiple-data stream computer is difficult and in some cases it is not possible, for example when subdomains are not equal in size or shape. Another type of parallel machine is a combination of different workstations through some software linkage, for example Parallel Virtual Machines.

The important factors to be considered in parallelization are:

1. Load balancing
2. Communication efficiency
3. Interface treatment

Equal load on all processors will give good performance on a parallel computer. This is a difficult task and maintaining this is cumbersome in some cases because of unequal domains. Communication efficiency depends on the number of interface nodes. Hence, interface treatment which depends upon the algorithm used for interface treatment is crucial for improving parallelization efficiency.

# CHAPTER 7

# CONCLUSIONS AND SCOPE FOR FUTURE WORK

## 7.1 CONCLUSIONS

The performance of operator splitting algorithm was tested in simple as well as complex geometries against the Bubnov-Galerkin method. Results were compared against the analytical solution of problems which were available in some cases. It was found that for low Peclet number flows, the performance of operator-splitting was identical to performance of other methods. For high Peclet number cases only the results by operator splitting came close to the analytical solution. Universal limiters were implemented for removing interpolation oscillations arising in high Peclet number flows. Operator splitting with universal limiters is relatively free from the upwinding errors and successfully simulates flows at high Peclet numbers as well as in complex geometries.

The performance of domain decomposition was tested in simple as well as complex geometries against the full domain problem. Results were compared with analytical solution which were available. The performance of domain decomposition depends mostly on the orientation of the interfaces, type of physical problem and interface treatment. This is clear from 2-domain, 4-domain, 9-domain problems with no cylinder and a single cylinder. The speed of domain decomposition depends on interface treatment and convergence criterion. For a convergence less than $10^{-4}$, there is no improvement in the results and only the number of iterations increases. With the help of domain decomposition we have tested the effect of outflow boundary on heat transfer from a cylinder placed in flow.

## 7.2 SCOPE OF FUTURE WORK

The following issues must be addressed in the future.

1. The performance of operator splitting and domain

decomposition for Navier-Stokes equations must be studied.

2. Uzawa's algorithm should be compared with conjugate gradient method for interface treatment to check for possible improvement.
3. The computer code developed in the present study must be implemented on a parallel machine to study improvements in CPU time.
4. Sensitivity of the solution to various meshes must be examined.
5. Simulations at very high Peclet numbers must be carried out.

Brooks, N. and Hughes, J.R., *Streamline upwind/Petrov-Galerkin Formulations for Convection Dominated Flows with Particular Emphasis on the Incompressible Navier-Stokes Equations*, Computer Methods in Applied Mechanics and Engineering, Vol. 32, pp. 199-259, 1982.

Cooke, C.H., *An Operator Splitting for Unsteady Boundary Value Problems*, Journal of Computational Physics, Vol. 67, pp 472-478, 1986.

Debasish Mishra, Muralidhar, K. and Ghoshdastidar, P.S., *Computation of Flow and Heat Transfer Around a Vertical Discrete Protruding Heater Using an Operator-Splitting Algorithm*, Numerical Heat-Transfer, Part A, 1995 (to appear).

Ding, Daoyang, and Liu Philip L., *An Operator Splitting algorithm for Two-Dimensional Convection-Dispersion-Reaction Problems*, International Journal for Numerical Methods in Engineering, Vol. 28, pp. 1023-11040, 1989.

Duff, I.S., MA28 - *A set of Fortran Subroutines for Sparse Unsymmetric Linear Equations*, AERE, Haswell, U.K., 1980.

Farhat and Lesoinne, M., *Mesh Partitioning Algorithms for the Parallel Solution of Partial Differential Equations*, Applied Numerical Mathematics, Vol. 12, pp. 443-457, 1993.

Ganagi, M.S., Singh, K.P., Athithan, G. and Atre, M.V., A *Fluid-Dynamics Study on ANURAG's Parallel Computer*, PACE-8, Current Science, Vol. 60, 1991.

Glaister, P., *An Approximate Linearised Riemann Solver for the Three Dimensional Euler Equations for Real Gases Using Operator Splitting*, Journal of Computational Physics, Vol. 77, pp. 361-383, 1988.

Glowinksi, R., Dinh, Q.V. and Periaux, J., *Domain Decomposition Methods for Non-Linear Problems in Fluid Dynamics*, Computational Methods in Applied Mechanics and Engineering, Vol. 40, 1983.

Glowinksi, R., Olivier Pironneau, *Finite Element Methods for Navier-Stokes Equations*, Annual Review of Fluid Mechanics, Vol. 24, pp. 167-204, 1992.

Issa, R.I., *Solution of the Implicitly Discretized Fluid Flow Equation by Operator Splitting*, Journal of Computational Physics, Vol. 62, pp. 40-65, 1985.

Laura, C., Dutto and Nagdi, G., Habashi, A Method for *Finite Element Parallel Viscous Compressible Flow Calculations*, International Journal for Numerical Methods in Fluids, Vol. 19,

pp. 275-294, 1994.

Leonard, B.P. and Simin Mokhtari, *Beyond first-order upwinding: The ultra-sharp alternative for non-oscillatory steady-state simulation of convection*, International Journal for numerical methods in Engineering, Vol. 30, pp. 729-766, 1990.

Le Tallec, P., *Domain Decomposition Methods in Computational Mechanics*, Computational Mechanics, Advances, IACM, Vol. 2, pp. 123-217, 1994.

Muralidhar, K., Sundararajan, T.,(Editors), *Computational Fluid Flow and Heat Transfer*, Narosa Publishers (in press), 1995.

Muralidhar, K., Varghese, M., Pillai, K.M., *Application of an Operator Splitting Algorithm for Advection-Diffusion Problems*, Numerical Heat Transfer, Vol. 23, pp. 99-114, 1993.

Pillai, K.M., Muralidhar, K., *Numerical Study of Oil Recovery Using Water Injection Method*, Numerical Heat Transfer, Vol. 24, pp. 305-322, 1993.

Reddy, J.N. and Gartling, D.K., *The Finite Element Method in Heat Transfer and Fluid Dynamics*, CRC Press, 1994.

Runnels, S.R. and Carry, G.F., *A Domain-Decomposition Strategy for Finite Element Simulation of Phase Change*. Numerical Heat Transfer, Part B., Vol. 24, pp. 181-189, 1993.

Sampaio, P.A.B., A Petrov-Galerkin/Modified *Operator Formulation for Convection-Diffusion Problems*, International Journal for Numerical Methods in Engineering, vol. 30, pp. 331-347, 1990.

Westerink, J.J., Shea, D., *Consistent Higher Degree Petrov-Galerkin Methods for the Solution of the Transient Convection-Diffusion Equation*, International Journal for Numerical Methods in Engineering, Vol. 28, pp. 1077-1101, 1989.

Yagawa, G., Soneda, N. and Yoshimura, S., *A Large Scale Finite Element Analysis Using Domain Decomposition Method on Parallel Computer*, Computers and Structures, Vol. 38, 1991.

Yanenko, N.N. and Shokin, Yu. I., *Numerical Methods in Fluid Dynamics*, MIR Publishers, Moscow, 1984.

# APPENDIX A

## FINITE ELEMENT METHOD

The finite element method is a powerful computational technique for the solution of differential and integral equations that arise in various fields of engineering and applied science. The finite element method is particularly advantageous for complex geometries and gradient boundary conditions especially when compared to the finite difference method. The major steps in finite element method applied to a typical problem are as follows:

1. Discretization of the domain into a set of finite elements (mesh generation).
2. Weighted - integral or weak formulation of the differential equation to be analysed.
3. Development of the finite element model of the problem using its weighted integral or weak form.
4. Assembly of finite elements to obtain the global system of algebraic equations.
5. Imposition of boundary conditions.
6. Simultaneous solution of the algebraic equations.
7. Post-computation of solution and quantities of interest.

Steps

1. Mesh Generation

In the present work, six-noded isoparametric triangular elements are used for accommodating the cylinder configuration correctly. Elliptic grid generation procedure is used for smoothening grids.

2. Weighted Integral or Weak Formulation

The governing differential equation is

$$T_t + u.\nabla T = \frac{1}{Pe} \nabla^2 T \qquad (A.1)$$

where $T_t$ is time derivative

$$\nabla \text{ is } \hat{i}\frac{\partial}{\partial u} + \hat{i}\frac{\partial}{\partial y}$$

$$\nabla^2 \text{ is } \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$

The symbols T, u are temperature and velocity respectively. To develop the weak form, we multiply Equation (A.1) by weight function w and integrate the equation over the element domain $\Omega^e$ $\Omega e$. Hence,

$$\int_{\Omega e} [T_t + u.\nabla T - \frac{1}{Pe}\nabla^2 T]\, W\, dX = 0 \qquad (A.2)$$

where $dX = [dx\ dy]$

After simplification of Equation (A.2) we get

$$\int_{\Omega e} [T_t W + (U.\nabla T)\, W - \frac{1}{Pe}(\nabla W.\nabla T)]\, dX - \frac{1}{Pe}\oint_{\Gamma e} \frac{\partial T}{\partial n} W\, ds = 0 \qquad (A.3)$$

Equation (A.3) subjected to initial condition T = 0 at t = 0 and boundary conditions are $\frac{\partial T}{\partial n} + hT = q$ on $\Gamma e$ where $\Omega e$ is the element domain and $\Gamma e$ is the boundary of that domain.

4. The dependent variable T is approximated over a typical element $\Omega^e$ by the expression

$$T(x,y) \approx T^e(x,y) = \sum_{j=1}^{n} T_j^e(t)\, N_j^e(x,y) \qquad (A.4)$$

where j = 1, ..., n, n is number of nodes per element. Here $N_j^e$ is shape function and is function of x and y only, and $T_j^e$ is function of time. $T_j^e$ denotes the values of the function $T^e(t,x,y)$ at selected number of points called nodes.

After imposing boundary conditions in Equation (A.3), we get

$$\int_{\Omega e} [T_t W + (U.\nabla T)\, W - \frac{1}{Pe}(\nabla W.\nabla T)]\, dX + \frac{1}{Pe}\oint_{\Gamma e} h\, T\, W\, d\, s - \frac{1}{Pe}\oint_{\Gamma e} q\, W\, d\, s = 0 \qquad (A.5)$$

Substituting Equation (A.4) in (A.5) we get,

$$\int_{\Omega e} [T_t W + (u.\nabla \sum_{j=1}^{n} T_j^e N_j^e)\, W - \frac{1}{Pe}(\nabla W.\nabla \sum_{j=1}^{n} T_j^e N_j^e)]\, dX + \frac{1}{Pe}\oint_{\Gamma e} h \sum_{j=1}^{n} T_j^e N_j^e W ds - \frac{1}{Pe}\oint_{\Gamma e} q\, W\, ds = 0 \qquad (A.6)$$

This equation must hold for any weight function W. In Bubnov-Galerkin method weight functions are equal to shape functions i.e. $W = N_j^e$. Since we need n independent algebraic equations to solve for the n unknowns, $T_1^e$, $T_2^e$, ..., $T_n^e$. We choose n independent functions for W namely $W = N_1^e$, ..., $N_n^e$. For each W, we obtain an algebraic equation among $T_1^e$, ..., $T_n^e$. We label the algebraic equation resulting from substitution of $N_i^e$ for W into Equation (A.6) as the first algebraic equation. The i-th algebraic equation is obtained by substituting $W = N_i^e$ into Equation (A.6).

Equation (A.6) becomes,

$$\sum_{j=1}^{n} (M_{ij}^e \dot{T}_j^e + K_{ij}^e T_j^e) = Q_i^e \qquad (A.7)$$

where the coefficients $M_{ij}^e$, $K_{ij}^e$ and $Q_i^e$ are

$$M_{ij}^e = \int_{\Omega e} N_i^e N_j^e \, dX$$

$$K_{ij}^e = \int_{\Omega e} [(U.\nabla N_j^e) N_i^e - \frac{1}{Pe} (\nabla N_i^e . \nabla N_j^e)] \, dX + \frac{1}{Pe} \oint_{\Gamma e} h N_i^e N_j^e \, ds$$

$$Q_i^e = \frac{1}{Pe} \int_{\Gamma e} q N_i^e \, ds$$

In matrix notation, Equation (A.7) takes the form

$$[M^e][\dot{T}^e] + [K^e][T^e] = [Q^e] \qquad (A.8)$$

where the matrix $[M^e]$ is called as mass matrix and $[K^e]$ is called as stiffness matrix or conductivity matrix.

Implicit$\phi$ finite difference scheme has been used in the present work for temporal approximation. After assembly of all these element matrices, we get the global matrix. This can be inverted by using direct methods or indirect methods. A direct inverter using Gaussian elimination with full pivoting has been used in the present work.

The integrals appearing in the definitions of the matrix entries are computed using Gaussian-quadrature. We have used 7 Gauss points in six noded isoparametric elements. These are,

$x_g$: 0, 1, 0, 0, 0.5, 0.5, 0.3333

$y_g$: 0, 0, 1, 0.5, 0, 0.5, 0.3333

The associated weights used in present work are,

$$W = \frac{1}{40}, \frac{1}{40}, \frac{1}{40}, \frac{1}{15}, \frac{1}{15}, \frac{1}{15}, \frac{1}{4.444}$$

The quadratic shape functions defined over a triangle are listed below.

$$N_1 = L_1 (2L_1 - 1)$$

$$N_2 = L_2 (2L_2 - 1)$$

$$N_3 = L_3 (2L_3 - 1)$$

$$N_4 = 4L_1L_3$$

$$N_5 = 4L_1L_2$$

$$N_6 = 4L_2L_3$$

where,

$$L_1 = 1 - \xi - \eta$$

$$L_2 = \xi$$

$$L_3 = \eta$$

where $\xi$, $\eta$ are natural coordinates on a unit triangle.

# APPENDIX B

## METHOD OF CHARACTERISTICS FOR ADVECTION STEP IN OPERATOR-SPLITTING ALGORITHM

The advection step in the OS algorithm generates is a hyperbolic equation. Consider the advection step of Equation (2.1),

$$T_t + u.\ \nabla T = 0 \qquad (B.1)$$

In full form the above equation is

$$T_t + u\ T_x + v\ T_y = 0$$

The above equation is simplified using streamline coordinates. We choose a curvilinear coordinate system ($\psi$, $-\phi$) locally at each node such that $\nabla\ \psi$ is aligned with the net composite velocity at that node, i.e. $\psi$ is the streamline passing through the node [See Figure B.1]. Functions $\psi$ and $\phi$ are orthogonal to each other. Then, it can be shown that

$$u\ T_x + v\ T_y = \bar{U}\ T_\psi \quad \text{where } \bar{U} = \sqrt{u^2 + v^2}$$

and $$T_\psi = \frac{\partial T}{\partial \psi}$$

Equation (B.1) now reduces to,

$$T_t + \bar{U}\ T_\psi = 0 \qquad (B.2)$$

The solution of this equation is

$$T\ (\ \bar{U}t - \psi\ ) = \text{constant} \qquad (B.3)$$

Since the grid surrounding each node is small, we assume the streamline $\psi$ to be locally straight over an element.

The solution of the corrector step requires that the temperature at 'Q' at n-th time step be transferred to the point 'P' at the (n+1)-th time step (see Figure B.1) i.e.,

$$T_P^{n+1}\ (\ \bar{U}(t + \Delta t) - \psi) = T_Q^n\ (\ \bar{U}t - (\ \psi - \Delta\psi)) \qquad (B.4)$$

where $\Delta\psi = \bar{U}.\Delta t$

If the advection step is solved after the diffusion step, $T^n$ refers to the solution obtained at the end of calculation of diffusion step. Here for finding the temperature value at 'Q' we use second order Lagrangian interpolation. The local master element coordinates $\xi$, $\eta$ are found from global x and y coordinates by using isoparametric relationship between $\xi$, $\eta$ and x, y. These isoparametric relations are used for mapping the original domain in the physical plane to a unit triangle. These relationships are

$$x = \sum_{j=1}^{n} x_j N_j (\xi, \eta) \qquad (B.5)$$

$$y = \sum_{j=1}^{n} y_j N_j (\xi, \eta) \qquad (B.6)$$

where n is number of nodes per element, $N_j$ are shape functions and are functions of $\xi$ and $\eta$. $x_j$ and $y_j$ are coordinates of nodal points. Equations (B.5) and (B.6) are solved using Newton-Raphson method (See Appendix E) to determine $\xi$ and $\eta$ as solutions of Equations (B.5) and (B.6).

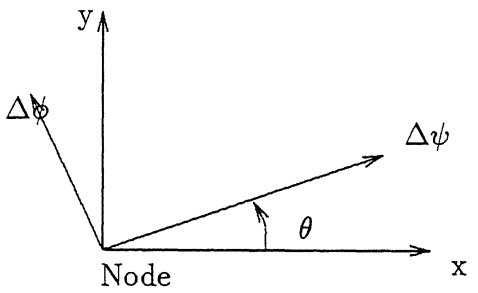



Figure B.1 Streamline Passing Through the Node.

# APPENDIX C

## ANALYTICAL SOLUTION

**ANALYTICAL SOLUTION FOR EQUATION (2.1)**

The advection-diffusion equation in one dimensional form with unit velocity is given by

$$T_t + T_x = \frac{1}{Pe} T_{xx} \qquad (C.1)$$

subjected to initial condition: $t = 0$, $T = 0$ and the boundary conditions:

$$x = 0, \quad T = 1$$

$$x \longrightarrow \infty, \quad T = 0$$

This represents a physical problem in which a hot fluid with unit velocity is introduced suddenly (at $t = 0$) at the inlet of a flow domain. The fluid in the physical domain ($0 < x < \infty$) is initially at a constant cold temperature. If the Peclet number is large, a thermal front whose thickness is small will move through the flow region with unit velocity. As $Pe \longrightarrow 0$, the front will be diffuse and energy transport is due to the molecular action of thermal diffusivity.

The closed-form solution of the above equation is,

$$T(x,t) = \frac{2}{\sqrt{\pi}} \int_{\sqrt{\frac{4t}{Pe}}}^{\infty} e^{-\eta^2} \, e^{Pe \left[ \frac{x}{2} \left( x^2 Pe / 16 \, \eta^2 \right) \right]} \, d\eta$$

where $\eta = \sqrt{4t/Pe}$ .

The above integral is well behaved and is evaluated using simpson's rule for numerical integration using 301 points.

# APPENDIX D

## BOUNDARY LAYER SOLUTION

Steady flow past a single cylinder at high Peclet numbers (Pe » 1) will lead to the formation of boundary layer on its surface. This is shown in Figure (D. 1). All changes in temperature are confined to the region $1 < r < 1+\delta$ and is zero outside this region. The boundary layer thickness shows on the cylinder surface and is a function of '$\theta$'. If the boundary layer is thin i.e. $\delta/R \ll 1$, where $\delta$ is the thermal boundary layer thickness, the governing equation for T can be simplified. If $\delta/R \ll 1$ the derivatives parallel to the cylinder wall are small in comparison to those normal to it i.e. $\frac{\partial}{\partial\theta} \ll \frac{\partial}{\partial r}$ and $\frac{\partial^2}{\partial\theta^2} \ll \frac{\partial^2}{\partial r^2}$. The reduced governing equations are,

$$(uT)_\theta = \frac{1}{Pe} ( rT_r)_r \qquad (D.1)$$

where r and $\theta$ are polar coordinates measured from the centre of the cylinder and u is now the tangential component of velocity. From potential theory the tangential conponent of velocity on the cylinder surface is $u = - 2 \sin\theta$ for flow past a single cylinder.

Equation (D.1) is solved by first assuming a profile for T in terms of r with $\delta$ as a parameter. Boundary layer thickness $\delta$ is then determined from an integrated form of Equation (D.1).

The form for T assumed in the present work is

$$T = \frac{\delta}{2} ( 1 - ( \frac{r-1}{\delta} ))^2$$

It satisfies the boundary conditions $r = 1$, $-T_r = 1$ and $r = 1+\delta$, $T = 0$ automatically. The value of T at cylinder surface is $\delta/2$. The equation governing '$\delta$' is obtained by integrating Equation (D.1) as $\int_1^{1+\delta} dr$. After integrating we get the following first order ordinary differential equation,

$$\frac{d\phi}{d\theta} + \phi \cot\theta = \frac{-3 \operatorname{cosec}\theta}{Pe} \qquad (D.2)$$

where $\phi = \delta^2$.

The initial condition for $\phi$ is obtained from Equation (D.2) by setting $\frac{d\phi}{d\theta} = 0$ as $\theta \longrightarrow \pi$.

Hence $\theta \longrightarrow \pi$, $\phi = 3/Pe$ and

$$\delta = \sqrt{3/Pe}$$

Equation (D.2) is solved by a fourth order Runge-Kutta scheme. Equation (D.2) exhibits singularity at $\theta = 0$ and $\delta$ and T are unbounded at this location. This point is excluded while determining the average Nusselt number of the cylinder by Simpson's rule.

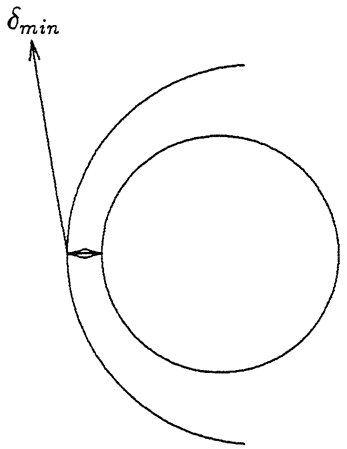



Figure D.1 Thermal Boundary Layer on the Surface of the Cylinder.

# APPENDIX E

## NEWTON-RAPHSON SCHEME FOR DETERMINING ($\xi$, $\eta$) FROM (X,Y) WITHIN A TRIANGULAR ELEMENT

The advection step requires transfer of temperature data from an interior point Q of one element to one of its nodes P (Appendix B). Figure (E.1) shows the isoparametric 9-noded Lagrangian master element. For finding ($\xi$, $\eta$) of point Q, the pair of simultaneous equations,

$$x_Q - \sum_{j=1}^{9} x_j N_j = f^Q(\xi, \eta) = 0 \qquad \text{(E.1)}$$

$$y_Q - \sum_{j=1}^{9} y_j N_j = g^Q(\xi, \eta) = 0 \qquad \text{(E.2)}$$

must be solved. According to Newton-Raphson scheme, we have to compute the corrections $\Delta\xi^Q$, and $\Delta\eta^Q$ to an initial guess using the following expressions.

$$\Delta\xi^Q = \frac{(f^Q g_\eta^Q - g^Q f_\eta^Q)}{\det}$$

$$\Delta\eta^Q = \frac{(g^Q f_\xi^Q - g_\xi^Q f^Q)}{\det}$$

where $\det = ( g_\xi^Q f_\eta^Q - f_\xi^Q g_\eta^Q )$

$$g_\xi^Q = \frac{\partial g^Q}{\partial \xi}, \text{ and } f_\xi^Q = \frac{\partial f^Q}{\partial \xi}$$

Here we know the $x_Q$, $y_Q$, $x_j$ and $y_j$. We assume ($\xi$, $\eta$) coordinates at point Q and calculate $\Delta\xi^Q$ and $\Delta\eta^Q$. If the calculated values of $\Delta\xi^Q$ and $\Delta\eta^Q$ are small, then the guessed ($\xi$, $\eta$) coordinates are correct, otherwise we modify $\xi$ and $\eta$ using following expressions.

$$\xi_{new}^Q = \xi_{old}^Q + \Delta\xi^Q$$
$$\eta_{new}^Q = \eta_{old}^Q + \Delta\eta^Q .$$

This procedure is repeated till convergence is reached.

The shape functions used for locating point Q are associated with a 9-noded Lagrangian element and are given as follows.

$$N_1 = \frac{1}{4} (\xi - 1) (\eta - 1) \xi \eta$$

$$N_2 = \frac{1}{2} (1 - \xi^2) (\eta - 1) \eta$$

$$N_3 = \frac{1}{4} (\xi + 1) (\eta + 1) \xi \eta$$

$$N_4 = \frac{1}{2} (\xi - 1) (1 - \eta^2) \xi$$

$$N_5 = (1 - \xi^2) (1 - \eta^2)$$

$$N_6 = \frac{1}{2} (\xi + 1) (1 - \eta^2) \xi$$

$$N_7 = \frac{1}{4} (\xi - 1) (\eta^2 + 1) \xi \eta$$

$$N_8 = \frac{1}{2} (1 - \xi^2) (\eta + 1) \eta$$

$$N_9 = \frac{1}{4} (\xi + 1) (\eta + 1) \xi \eta$$

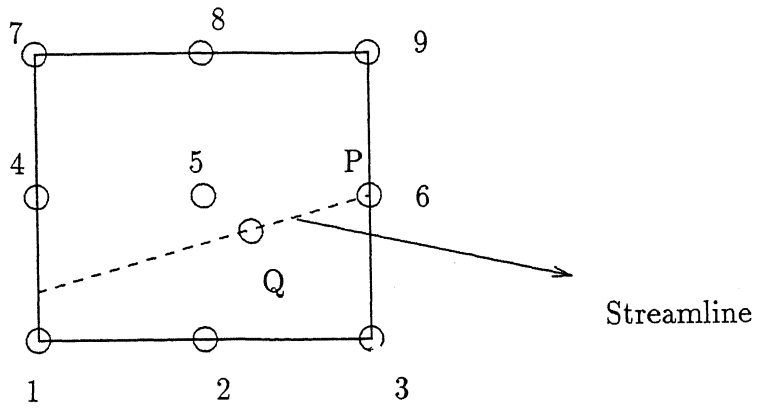



Figure E.1 9 Noded Isoparametric Lagrangian Master Element.